Blender » Scribus » C# » Джереми Эллисон

LXF DVD



StartCom 5
Новый супернадежный дистрибутив с Compiz

» 80 решенных проблем
» OpenOffice.org 2.2

LINUX FORMAT

Главное в мире Linux

Июнь 2007 № 6 (93)



Властелин Сетей

Навыки, необходимые администраторам всех мастей:

» Блокирование нарушителей
» Совместное использование файлов
» Скоростная загрузка » И даже больше!

Ubuntu 7.04

Feisty Fawn выскочил на опушку с. 10

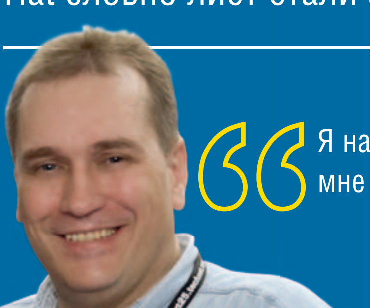
42 страницы разнообразных учебников на любой вкус »

Внедряйте SELinux
Система безопасности от Red Hat словно лист стали с. 46

Познакомьтесь с D
C++ со сборкой мусора или C# без виртуальной машины с. 50

Кодируйте с Nvu
Вся мощь web-приложений – и никаких проблем с. 66

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974



«Я нахожусь в таком месте, где полно людей, и каждый мне говорит: «Ты – идиот, всё надо делать не так!»»

Джереми Эллисон теперь работает в Google с. 34

LINUX FORMAT
Главное в мире Linux

К Вашим услугам...

Примите горячие приветствия от команды LXF, которая в этом месяце была озадачена вопросом: «Какой, по-вашему, самый подходящий способ отблагодарить системного администратора за его нелегкий труд?»



Пол Хадсон
Поменяться на денек местами с директором. Наслушавшись ноющих пользователей, тот бы выдал ему служебную Ferrari – за вредность.



Грэм Моррисон
Сводить его к стилисту и подарить бесплатную подписку на услуги сайта знакомств.



Майк Сондерс
Что-нибудь новенькое, игрушка и шоколад. Да, это он – гигантский киндер-сюрприз с Nintendo DS, бухтой Ethernet-кабеля и котен-ком-роботом внутри!



Эфрейн Эрнандес-Мендоса
Выплатить месячное жалование и отправить на курсы по обучению администрированию Windows Vista.



Ребекка Смолли
Уверена, звонка от меня в стиле «Извините, я воткнула клавиатуру и теперь все работает» будет достаточно.



Нейл Ботвик
Сисадминам не нужна награда. Им власть дана в любовницы, мне, право, очень грустно...



Дэвид Картрайт
Ввести всемирный запрет на фразы: «Что-то сеть барахлит» и «У меня срочный отчет – подойди немедленно».



Энди Ченнел
Нужно заставить их участвовать в реалити-шоу типа Последний герой, отправить на необитаемый остров, а потом кого-нибудь зажарить и съесть.



Эмиас Ченнер
Необходимо человеческое жертвоприношение. Каждый администратор должен выбрать кандидата, принесшего ему больше всего ненужной работы.



Бен Харлинг
Выдать лицензию на убийство, как у Бонда и вагон кукол вуду для защиты от предложений о реструктуризации впридачу.



Энди Хадсон
Организовать путешествие в какую-нибудь жаркую страну – за счет заведения. Им не достает солнца, и это будет неплохой шанс поправить здоровье.



Рейчел Проберт
Денек на солнышке, с бутылочкой солнцезащитного крема фактора 50 или выше и панамкой... или это слишком жестоко?

Миссия журнала

Corporate Patron — Free Software Foundation

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО



Встречаем альтернативные ОС?

Нет, я не собираюсь злоупотреблять отведенным мне местом, чтобы лишний раз упомянуть Syllable, Haiku, ReactOS – и их также нет на нашем DVD. Решения, о которых пойдет речь, [пока что] базируются на Linux – но являются альтернативными совсем не поэтому.

Компоновка по схеме «компьютер – жесткий диск – операционная система – установленные приложения» настолько прочно вошла в нашу жизнь, что впору задуматься, а может ли быть иначе. И последние разработки в мире Open Source решительно утверждают – может.

В первую очередь, на ум приходит Mandriva Flash – полноценная ОС, запускающаяся с USB-брелка. Сама идея не слишком нова, но оригинальное исполнение вкупе со стремительно дешевеющими носителями большой емкости открывает для нее интересные перспективы. Кто знает, может быть, спустя несколько лет в крупных аэропортах появятся стойки с мониторами и табличками: «Just plug your USB stick here and start working»?

Другое перспективное направление – работа через Интернет. Это тот самый виртуальный сетевой компьютер, о котором незабвенный Билл Гейтс говорил еще 10 лет назад. Сегодня идея помаленьку обретает форму, но кого мы видим на передовой? Red Hat Global Desktop и Ulteo, детище Гаэля Дюваля.

Природа Open Source подстегивает инновации, и потому будущее выглядит для нас светлым. **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru

› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.126

Июнь 2007 № 6 (93)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники

Система
Все эти каталоги 58
Для вас, новички: /home, /root, /bin – и другие места в файловой системе, имена которых могут смутить нормального человека.

Hamachi
Безопасное соединение 62
Виртуальные частные сети открывают безграничные возможности. Воспользуйтесь нашим пошаговым руководством, чтобы создать себе такую с Hamachi.

Web
Менеджер коллекций 66
Какая из моих записей хоральной прелюдии Vater unser im Himmelreich Баха самая старая? Минутку, мне нужно подключиться к Интернету...

Mono
Объектно-ориентированная игра 70
Мы займемся ООП, не ООН, хотя результат будет сходным – и в том, и в другом случае можно будет попробовать поуправлять миром.

Blender
Термерный чат 74
Занимательный проект для бородатых хакеров: трехмерный чат, полезный сам по себе или в паре с какой-нибудь MMORPG.

GTK+
Текстовый редактор 78
Интегрированная среда с красивым именем Anjuta значительно упрощает процесс разработки, поэтому мы можем взяться за проект посерьезнее.

Java EE
Доступ к СУБД 82
[Ныне открытая] платформа Java предлагает унифицированный интерфейс к любой базе данных – осталось только научиться им пользоваться.

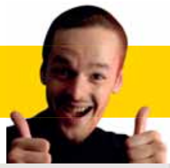
LaTeX
Табличная верстка 86
Мы не будем рассуждать о преимуществах DIV и CSS – хотя бы потому, что они не имеют никакого отношения к теме статьи. Вместо этого, мы покажем как создавать в LaTeX красивые и полезные таблицы.

Электронные подписи
Цифровые сертификаты 90
Лучший способ подтвердить свою личность в Сети – это цифровой сертификат: современная криптография на страже интересов рядовых пользователей Linux.

Cinelerra
Маски и проекторы 94
Изучите основные подходы к применению открытого нелинейного видеоредактора для решения практических задач.

Mencoder
Ужимаем видео 97
Десятиминутное руководство по конвертированию мультимедиа-файлов.





LXF DVD93

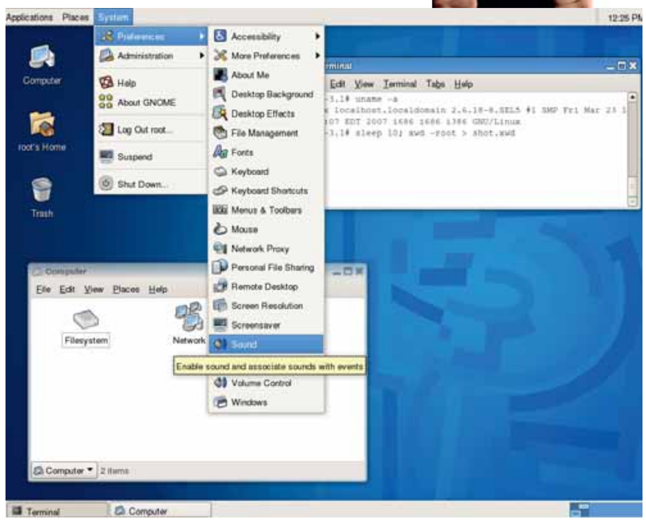
Майк вам покажет 108

StartCom EL 5
Ищите надежный, хорошо протестированный дистрибутив для дома или офиса? Познакомьтесь со StartCom – клоном Red Hat Enterprise Linux без ежемесячных отчислений.

OpenOffice.org 2.2
Флагманский открытый пакет офисных приложений стал еще более совместимым с MSO и, что лучше всего, более скорым на подъем! Правда, ненамного.

Журнал в PDF
Три статьи из предыдущих выпусков LXF. Узнайте, как собрать собственный Linux-киоск, разберитесь с лицензией Creative Commons и найдите неожиданное применение для Subversion.

Игры
Космическая стрелялка ChouXiang, морские приключения с FunnyBoat и пища для размышлений с PyTraffic.



› StartCom LInux: этот фантастически настольный дистрибутив.

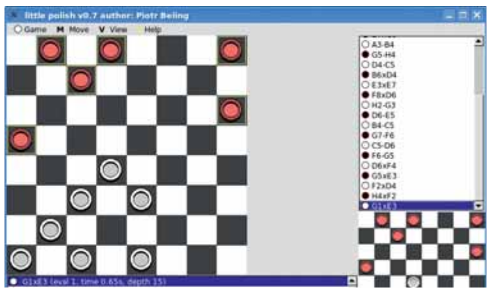
Что за штука...

D? C++ становится лучше. «Все флаги в гости будут к нам» с. 50



LXFHotPicks

Лучшие новинки открытого ПО на планете 116



› Ты потопил мой четырехпалубник! Ой, это из другой оперы...



LXF ИНТЕРВЬЮ

«Я нахожусь в таком месте, где полно людей, и каждый мне говорит: «Ты – идиот, всё надо делать не так!»

Джереми Эллисон с. 34



Время творить! с. 38
Студия Эмиаса Ченнера переезжает в Linux



Подпишись на Linux Format и сэкономь!

Содержание



LXF DVD внутри!

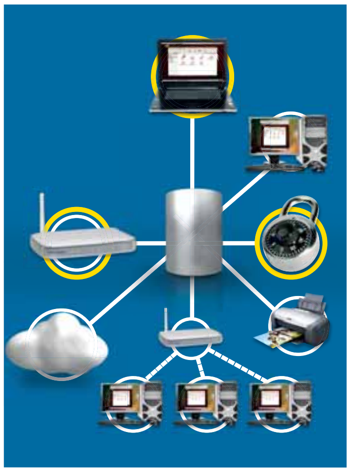
См. страницу 108

Спецрепортаж

Арсенал сисадмина

Как упростить и ускорить управление вашей системой с. 26

А также...



Мытарства композитора 38
Создаем Linux-студию звукозаписи.

KDE4 на колесах............................ 42
Мгновенный перенос пользователей с Kamion: предварительный обзор и HOWTO.

Доступно о SELinux 46
Проверенные подходы к укреплению безопасности

Инструменты мобилизации........... 52
Подключите свой сотовый телефон к компьютеру, чтобы использовать его на полную катушку

Постоянные рубрики

Новости.............................. 04

DistroWatch........................ 24
Ладислав Боднар делится впечатлениями от Ubuntu 7.04 и утверждает, что Mandriva таки-нашла ключ к зарабатыванию денег. Или хотя бы «брелок» от него...

Интервью LXF.................... 34
Чем теперь займется Джереми? Мы бы хотели увидеть его в Танцах со звездами, а он вместо этого будет работать над Samba в Google...

Что за штука.... 50
Еще один язык программирования для истинных ценителей: встречайте логическое развитие C++!

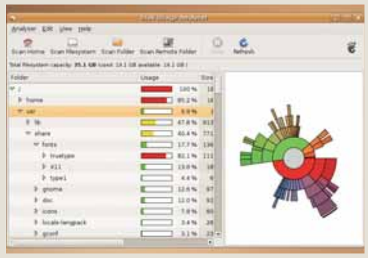
Ответы 100
В этом месяце: модемы, ./configure, QEMU, SSH и другие ваши проблемы, а также настройка беспроводного соединения!

Через месяц 126
Что ждет нас в июле?



Обзоры

Ubuntu 7.04 10
Пан или пропал? Энди Хадсон засиделся на зеркалах Ubuntu, пытаясь узнать, как были решены проблемы с проприетарными драйверами.



› Feisty Fawn: выпутавшийся из уз заботы Dapper Drake?

PuppyLinux 2.16 12
Оригинальный LiveCD с приятным логотипом и букетом интересных возможностей. Где бы раздобыть для него старый компьютер?

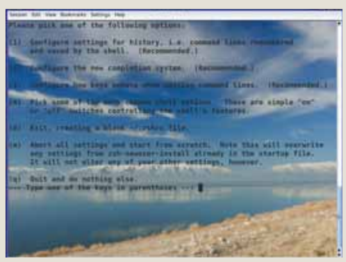
Scribus 1.3.3.8 14
Поддержка графиков, улучшенный GUI, управление цветом – и даже больше...

Squeezebox 3 15
Супер-крутой довесок к вашей Hi-Fi системе. Кто сказал: «Секундочку, я запущу браузер»?

Storix SBAdmin 6.1 16
Резервирование из окна любого браузера – достойный соперник NetVault в сегменте малого бизнеса.

Сравнение: командные оболочки

Bash **19**
Fish **20**
Tcsh **20**
Zsh **21**
Dash **21**
Dsh **22**
Sash **22**



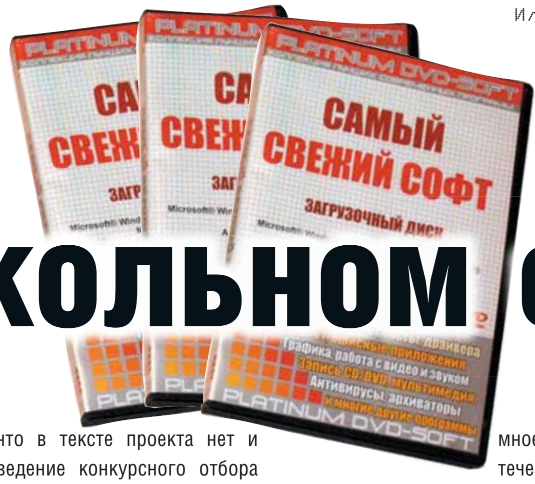
LINUX FORMAT

Новости

ГЛАВНЫЕ НОВОСТИ: » Школьный софт » GPL v2 на свободной лицензии » Red Hat Global Desktop » SandLinux » Психическая атака » Ubuntu для Microsoft » О Linux

» Рубрику ведет Илья Шпаньков





Сага о школьном софте

В конце апреля Министерство информационных технологий и связи РФ представило на рассмотрение план действий по обеспечению российских школ легальным ПО. Дело это, безусловно, полезное и нужное, но ознакомление широкой общественности с подробностями предлагаемой программы вызвало у сторонних наблюдателей неоднозначную реакцию, начиная от недоумения и заканчивая откровенным протестом. Суть предложения заключается в том, чтобы провести централизованную закупку необходимого ПО для всех имеющихся в школах компьютеров, общее число которых составляет около 670 тысяч. За счёт объёмности заказа Мининформсвязи планировало получить от поставщиков значительные скидки и возможность рассрочки платежа, что позволило бы снизить затраты на покупку необходимого программного обеспечения до 787 рублей в год в расчёте на один компьютер. В общей сложности из федерального бюджета предлагалось потратить около 3,8 миллиарда рублей в течении пяти ближайших лет. Собственно цифры никого не удивили – на хорошее дело можно и раскошелиться, но вот список предполагаемых к покупке программных продуктов, включающий не только операционную систему Windows, но и профессиональные пакеты для работы с графикой Adobe Photoshop и Corel Draw, а также несколько других спорных позиций, был раскритикован полностью.

Первыми высказали недоверие данной программе представители ведомства Германа Грефа, усмотревшие в ней не только очевидные недочёты, но и явные нарушения действующего в России на данный момент антимонопольного законодательства. Помимо того, что в тексте проекта нет и намёка на проведение конкурсного отбора ПО среди нескольких претендентов, закупка исключительно проприетарного программного обеспечения также рассматривается чиновниками Минэкономразвития как недопустимая. Свободные аналоги закрытых программ, получившие сегодня довольно широкое распространение, практически ни в чём не уступают дорогостоящим «коллегам», при этом многие из них являются платформенно-независимыми и отлично работают в среде Windows. Между тем, в предложении Мининформсвязи ни слова не говорится о возможности использования свободного ПО. По мнению Минэкономразвития, проект в том виде, в котором его представил руководитель Мининформсвязи Леонид Рейман, не может быть принят в работу и требует основательной корректировки. Свою лепту в обсуждение актуальной темы внесли и финансисты: в начале мая Министерство финансов РФ отправило в Мининформсвязи письмо, в котором отказалось выделять деньги из бюджета на реализацию проекта. Впрочем, основной причиной данного решения стало, скорее, желание перенести бремя ежегодных платежей на плечи региональных органов управления. Никаких претензий к собственно списку ПО финансовые чиновники не высказали.

Чуть позже директор департамента Мининформсвязи Максут Шадаев попытался объяснить позицию своего ведомства и, справедливости ради, нужно отметить, что многие доводы выглядели вполне логично. В целях скорейшего решения проблемы нелицензионного ПО в школах министерство пошло самым простым путём, предложив легализовать уже используемое в учебных заведениях программное обеспечение. Кроме того, вот уже в течении нескольких лет для школ разрабатываются учебные приложения – как нетрудно догадаться, функционально привязанные к среде Windows. Таким образом, кардинальные изменения в данной сфере повлекут за собой и довольно ощутимые негативные последствия. В то же время чиновники не отрицают, что переход на использование свободного ПО в образовательной сфере не только возможен, но и необходим, поэтому одновременно с легализацией уже используемого проприетарного программного обеспечения планируется запустить пилотный проект на базе операционной системы GNU/Linux в одном из российских регионов.



Следует также отметить, что осторожный подход к процессу миграции на свободное ПО поддерживают и крупнейшие российские разработчики Linux-систем – компании AltLinux и ASPLinux. И их доводы не менее разумны: если сейчас школы в приказном порядке заставить переходить на совершенно незнакомое им программное обеспечение, то это может привести только к негативным последствиям и в итоге надолго закроет свободному ПО дорогу в школы и вузы. Для начала нужно как минимум создать дистрибутив, отвечающий всем требованиям министерства образования, затем обучить принципам работы с ним преподавательский состав, и только потом начинать широкое внедрение. Впрочем, первый этап уже практически завершён: компания AltLinux к сентябрю текущего года закончит подготовку специализированного дистрибутива, создаваемого в соответствии с рекомендациями Минобраза и полностью адаптированного для российских школ.

http://www.minsvyaz.ru/news/

Лучше поздно, чем никогда



В начале мая компания Sun Microsystems завершила процесс перевода Java на свободную лицензию GPL v2. Вдобавок к компонентам, переданным сообществу FOSS в конце прошлого года, новые условия лицензирования распространились на такие ключевые пакеты, как JDK 2 (Java Development Kit 2) для Java SE (Java Platform Standard Edition). Кроме того, пользователи OpenJDK получили возможность работать с JCK (Java SE 6 Technical Compatibility Kit) для обеспечения совместимости своих приложений со спецификацией Java SE 6. Таким образом, компания полностью открыла исходные тесты большинства компонентов технологии Java, что было с энтузиазмом встречено приверженцами данной платформы. По словам исполнительного вице-президента компании Рича Грина [Rich Green], «теперь свободное сообщество получило доступ ко всем вариантам Java-платформы – Standard, Enterprise и Micro Editions, распространяемым отныне по лицензии GPL». Также представители Sun с определённой долей гордости сообщили, что данный вклад в FOSS является крупнейшим из всех, когда-либо происходивших в индустрии ПО.

Впрочем, помимо благодарственных речей не обошлось и без вполне обоснованных упрёков. По мнению многих разработчиков, код Java нужно было открыть на несколько лет раньше, а еще лучше было бы изначально продвигать эту платформу в качестве свободной. В таком случае молодые конкуренты Java – C# и CLR от Microsoft – имели бы гораздо меньше шансов на успех, а само развитие мощного и удобного программного инструмента происходило бы гораздо быстрее. Остаётся надеяться, что в результате перевода Java на свободную лицензию данная платформа сможет более успешно противостоять аналогичным проприетарным разработкам.

Завтра наступает сегодня

Окружающий мир постоянно эволюционирует, и данный процесс наиболее явно прослеживается в области высоких технологий, где новые разработки уже к моменту появления на прилавках переходят в разряд «вчерашний день». Поэтому крупным разработчикам, желающим сохранять лидирующие позиции на рынке, всегда нужно быть на самом переднем крае инноваций, дабы встретить наступление новых технологий во всеоружии. Именно так поступает компания Red Hat, в начале мая объявившая о готовности нового необычного продукта – Red Hat Global Desktop (RHGD). По сути, это операционная система, но значительная часть её компонентов представляет из себя онлайновые сервисы поколения Web-2.0, тесно интегрированные с базовым набором системных приложений. Таким образом разработчики основательно разгружают устанавливаемую на компьютер ОС, одновременно снижая и аппаратные требования.



По идее авторов проекта, работать в RHGD можно будет на довольно слабых машинах (компания Intel также проявила интерес к новой системе, пожелав принять участие в разработке и опробовать её возможности на ультрадешёвом ноутбуке Classmate PC), тем самым обеспечив самыми современными технологиями большое число небогатых пользователей из развивающихся стран, которые не могут себе позволить покупку тысячедолларовой персоналки, пригодной для установки той же Windows Vista. Как справедливо заметил технический директор компании Red Hat Брайан Стивенс [Brian Stevens], «концепция стационарного компьютера уходит в прошлое – пришло время сетевых решений, позволяющих работать в привычном программном окружении с любого устройства и из любой точки мира». В перспективе новая концепция позволит использовать в качестве «тонкого клиента» даже карманные компьютеры или смартфоны, при этом пользователь будет по-прежнему иметь доступ ко всему функциональному богатству, свойственному сегодняшнему ПК. Остаётся только добавить, что первая версия RHGD должна быть обнародована в июне текущего года, а основой её послужит свободный дистрибутив Fedora.

http://www.redhat.com/about/news/prarchive/2007/global_desktop.html

Самый быстрый и реальный

Ежегодный женевский международный салон изобретений принёс российским разработчикам бронзовую медаль за представленную широкой публике универсальную среду разработки программного обеспечения для встраиваемых систем реального времени SandLIX. Авторами проекта являются компания Cyber System Development и Ульяновский государственный технический университет, а основой уникальной разработки послужила операционная система реального времени SandLinux, уже не первый год разрабатываемая компанией Cyber System Development. В комплект SandLIX входит четыре модуля: непосредственно ОСРВ SandLinux, автоматизированная система сборки образа ОСРВ для целевой платформы, система отладки приложений реального времени и набор тестов Test Suites SL. Российским программистам удалось добиться действительно отличных результатов: время загрузки данной операционной системы почти в 20 раз меньше, чем у аналогичных решений, для работы SandLinux требуется в 15 раз меньший объём оперативной памяти, при этом время отклика системы составляет всего 8 миллисекунд. Подобные решения обладают большим спросом в областях, где возможность работы в режиме реального времени является жизненно необходимой – это и авиация, и научные исследования, и, естественно, военная промышленность. В данный момент доступна версия операционной системы SandLinux 1.1.

www.cybsysdev.com



Психическая атака

Разговоры о том, что создатели и пользователи GNU/Linux систем нарушают патенты разработчиков коммерческого ПО, с определённой периодичностью возобновляются с подачи той или иной компании. Когда-то SCO Group потрясала в воздухе охапкой судебных исков, пытаясь привлечь свободное сообщество к ответственности за «украденный» код, но так и осталась ни с чем, не предъявив ни одного сколько-нибудь серьёзного доказательства. Новая волна нападок на FOSS инициирована главным производителем ПО – компанией Microsoft. Причём если в конце прошлого года представители редмондской корпорации обходились намёками, то нынешний май изобиловал более резкими выпадами, для пущей важности дополненными некоторыми цифрами. Так, юристы Microsoft сообщили журналистам, что операционные системы GNU/Linux нарушают 235 патентов гиганта индустрии, а чуть позже исполнительный директор Стив Баллмер [Steve Ballmer] увеличил это число до 283. Вывод у Microsoft напрашивается один – за использование интеллектуальной собственности нужно платить. Как это сделала, например, компания Novell, заключившая в прошлом году с редмондским гигантом договор о взаимном сотрудничестве, содержащий довольно туманный и неоднозначно принятый сообществом пункт о патентных урегулированиях. При этом пикантность ситуации в том, что Microsoft и Novell толкуют данный пункт каждый по-своему: Novell всех убеждает, что никакого признания нарушений патентов он не подразумевает, а Стив Баллмер при каждом удобном случае повторяет, что все выплаты Novell по договору являются исключительно патентными отчислениями.

Впрочем, разговоры пока по-прежнему остаются разговорами: никаких конкретных нарушений или примеров использования запатентованных Microsoft разработок не предъявлено, и многие аналитики уверены, что судебных исков не будет и в дальнейшем. Прежде всего это невыгодно самой Microsoft: при нынешней путанице в современной патентной системе доказать, что кто-то у кого-то что-то позаимствовал – дело довольно сложное и затратное. Кроме того, у свободного ПО сейчас появилось достаточно много влиятельных сторонников в лице той же IBM, которые обладают собственным огромным пакетом патентов и при необходимости легко могут инициировать ответный букет претензий. Есть и более любопытные доводы: основатель проекта Ubuntu Марк Шаттлворт [Mark Shuttleworth] вообще считает, что Microsoft и сообщество Linux-разработчиков находятся по одну сторону фронта, т.к. оба легко могут стать объектом патентных исков со стороны компаний, для которых сутяжничество на почве программного обеспечения – единственный источник доходов. А повод всегда найдётся: перефразируя известную поговорку, можно сказать, что «не нарушает патентов тот, кто ничего не производит». Таким образом, большинство сторонних наблюдателей склонно расценивать угрозы со стороны Microsoft лишь в качестве средства психологического воздействия,



причём целевой аудиторией является отнюдь не свободное сообщество, которое на подобные нападки давно выработало иммунитет, а огромная армия потенциальных пользователей новой версии Windows Vista, которые слабо разбираются в юридических тонкостях, но предпочитают лишний раз перестраховаться и не связываться со свободным ПО, о котором «говорят такое».

Стоит также отметить, что сами того не желая, маркетологи Microsoft таким образом могли запустить механизм замедленного действия, который в перспективе способен привести к значительным реформам патентного законодательства и потере редмондским гигантом такого удобного козыря: намучившись с объяснениями злосчастного пункта договора с Microsoft перед FOSS-сообществом, компания Novell совместно с фондом Electronic Frontier Foundation приступили к разработке совместных действий по лоббированию национальных и международных законотворческих организаций (вплоть до World Intellectual Property Organization - WIPO) на предмет кардинального пересмотра принципов существующей патентной системы, из инструмента научно-технического прогресса превратившейся в препятствие, не отвечающее современным реалиям софтверной эволюции. Если инициативу поддержат и другие влиятельные игроки рынка, то результат может оказаться довольно неприятным для сторонников «гонки патентных вооружений».

Ubuntu для Microsoft

Компания Dell, переживающая сейчас не лучшие времена, довольно часто стала появляться в заголовках новостей. Наибольший интерес, естественно, вызвало решение компании выпустить в продажу компьютеры с предустановленной операционной системой Ubuntu GNU/Linux, что вряд ли кто-то мог предположить ещё какие-то полгода назад. Между тем, уже сегодня в магазинах на территории США можно приобрести недорогой настольный ПК из бюджетной линейки Dimension E520, ноутбук Inspirion 1505 или компьютер уровня домашнего развлекательного центра XPS 410, укомплектованные свободной операционной системой. Но это только начало – в дальнейшем Dell планирует расширить линейку Linux-машин и устанавливаемого на них ПО, и наиболее вероятным кандидатом выступает операционная система SUSE Linux от компании Novell. Об этом говорит тот факт, что в мае Dell присоединилась к договору между Novell и Microsoft, в рамках которого редмондский гигант получил возможность предлагать своим клиентам пакеты лицензий на SUSE Linux. Теперь аналогичными правами обладает и Dell, планирующая продавать SUSE Linux Enterprise Server в комплекте с серверными системами собственного производства. При этом не исключено, что и Microsoft добавит в свой ассортимент продукцию Dell, тем самым предлагая клиентам комплексные аппаратно-программные решения. Как это ни парадоксально звучит, но в данной ситуации совершенно не исключена возможность того, что в дальнейшем редмондский гигант будет продавать помимо всего прочего и операционную систему Ubuntu, основанную на самом свободном в мире дистрибутиве Debian. Впрочем, вряд ли авторов Ubuntu ожидает судьба Novell, получившей немало обидных слов за свой договор с Microsoft: всё-таки разработчиком данной свободной операционной системы Dell не является.

ubuntu

linux for human beings



Реклама









Фото: Макс Борзов, www.internetmarafon.ru

В Петербурге снова говорили о Linux

На сей раз – в рамках масштабной Санкт-Петербургской Интернет Конференции. Она прошла в Ольгино 18 мая под патронажем РОЦИТ и собрала большое количество петербургских и областных деятелей интернет-бизнеса и ИТ.

Среди многочисленных секций, докладов и семинаров присутствующая публика не обошла вниманием и линуксоидов, а конкретнее – секцию «ПО на базе Linux и Open Source: решения для веб-серверов и рабочих станций».

Среди докладчиков особый интерес аудитории вызвал Павел Фролов, директор проекта «Линуксцентр», который рассказал, что делать предприятиям в свете усилившегося контроля государства за пиратским софтом, особенно в области образования.

Преимуществами свободно распространяемого программного обеспечения применительно к учреждениям школьного образования, по мнению Павла, являются меньшая стоимость его приобретения, поддержания и владения – возможность модифицирования, доработки и усовершенствования приобретенного ПО с целью наиболее полного соответствия специфическим нуждам преподавателей и учащихся, информационная безопасность компьютерных систем и данных,повышение производительности труда пользователей.

Юрий Васильев, заведующий лабораторией ГУАП, рассказал о преимуществах перехода на Linux крупных предприятий и организаций. В простой и доступной форме презентации было наглядно показано – на сегодняшний день все необходимые для функционирования ИТ-структуры предприятия звенья присутствуют не только у коммерческих поставщиков, но и в мире Open Source. Офис, корпоративные порталы, web-серверы, даже ERP- и CRM-системы – уже есть все, дело за внедрениями.

На «сладкое» Дмитрий Дмитриев из Линукс Инк поведал аудитории о портальной технологии Plone и о новых методах поиска в Интернете, способных в будущем перевернуть существующий рынок поисковых сервисов. Правда, в глубинную суть новых технологий могли вникнуть только специалисты.

Провел мероприятие Павел Житнюк, директор по развитию новой компании iTREND, миссией является популяризация Open Source и внедрение свободного ПО на предприятиях и организациях России.

Отрадно, что с каждой такой публичной акцией аудитории становятся все более заинтересованными в предмете обсуждения. Ситуация на ИТ-рынке складывается таким образом, что Linux входит в моду – и это радует. LXF

www.internetmarafon.ru



Новости короткой строкой

» Компания Microsoft объявила о новой инициативе под названием Student Innovation Suite – это пакет приложений, включающих Windows XP Starter Edition, Microsoft Office Home and Student 2007, Microsoft Math 3.0, Learning Essentials 2.0 для Microsoft Office и Windows Live Mail desktop, стоимость данного набора составит всего $3.

» Власти девятнадцати северо-американских штатов проявили интерес к проекту стодолларового ноутбука One Laptop per Child (OLPC), рассматривая возможность его распространения и в американских школах.

» 60-летний французский программист Мишель Ксаар [Michel Xhaard] за три года написал Linux-драйверы к 235 моделям USB web-камер.

» Представители итальянского офиса продаж компании Toshiba сообщили, что на всех ноутбуках серии Tecra, Satellite, Portege и Qosmio будет по желанию заказчиков предустанавливаться операционная система Linux.

» Компании Oracle Corp., IBM Corp. и NEC Corp. создали консорциум, основная цель которого – продвижение на японский рынок серверов и других аппаратных систем на базе Linux.

» 7 мая Верещагинский районный суд повторно рассмотрел «дело Поносова» и признал директора сельской школы Пермского края виновным, обязав его выплатить в пользу государства штраф в размере 5000 рублей.

» Компании Canonical и Intel объявили о начале совместного проекта Ubuntu Mobile and Embedded Edition в целях создания полнофункциональной аппаратно-программной платформы для использования в мобильных устройствах различного назначения.

» 7 мая объявлен долгожданный официальный релиз дистрибутива Gentoo Linux 2007.0.

» Компания Red Hat запустила онлайновый сервис Red Hat Exchange, на котором можно быстро оформить покупку популярных коммерческих приложений для среды Linux, и выпустила под свободной лицензией набор высококачественных шрифтов LIBERATION, являющихся аналогом входящих в состав Microsoft Windows.

LINUX FORMAT

Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Свою первую (и последнюю) программу написал еще на Алголе

Мир изменился...

И произошли эти изменения стремительно – на протяжении первых месяцев текущего года. Еще совсем недавно слова «Open Source» и «Linux» были нечастыми гостями даже на страницах общекомпьютерной прессы, выставки и конференции этой тематики, хотя и имели место быть в последние годы, собирали исключительно узкий круг профессионалов, а Linux-сайты и форумы, хотя и были традиционно многочисленными, посещались лишь энтузиастами (и теми, кто готовился таковыми стать).

Что же мы увидели за истекшие месяцы? Слова «Open Source» и «Linux» на страницах СМИ, по крайней мере онлайновых. Конференции и круглые столы, организуемые не только крупными информационными агентствами, но и политическими партиями. Депутат Думы, бывший советский полковник и диссидент, рассуждает о необходимости создания отечественной операционной системы. В ряды посетителей линуксфорумов, вслед за представителями технократической элиты и технологической богемы, влилась творческая интеллигенция гуманитарной линии.

Не суждено ли нам в ближайшие месяцы увидеть материалы по нашей тематике на страницах политических еженедельников, специализированных изданий для силовиков и любителей домоводства, рыболовов-спортсменов и автолюбителей? А то – и в «глянцевых» журналах для мужчин, женщин и сексуальных меньшинств.

Хорошо это или плохо для Linux и Open Source – покажет время. Однако можно тешить свое тщеславие – к текущему изменению мира руку приложили и мы...

alv@posix.ru

Сегодня мы рассматриваем...

10 Ubuntu 7.04
Edgy не был таким уж рисковым и Feisty [англ. храбрый] видимо пойдет тем же путем. Растерял ли Ubuntu свой боевой настрой, или мы настолько довольны им, что не хотим ничего улучшать? Попробуем разобраться.

12 Puppy Linux 2.16
Щенки на логотипе – писк этого сезона, но что еще может предложить концепция LiveCD придирчивому пользователю? **Сергей Супрунов** в очередной раз убедился, что собака – действительно друг человека.

14 Scribus 1.3.3.8
Быть может, это действительно лучшее решение для верстки документов, но готов ли Scribus нести свет Open Source в мир больших DTP? Мы решили посмотреть на него поближе...

15 Squeezebox 3
Грэм Моррисон уверен, что Squezzebox изменит смысл словосочетания «слушать музыку», но не слишком доволен ценой. Наверное, нам стоит платить ему больше. Или хотя бы просто – платить.

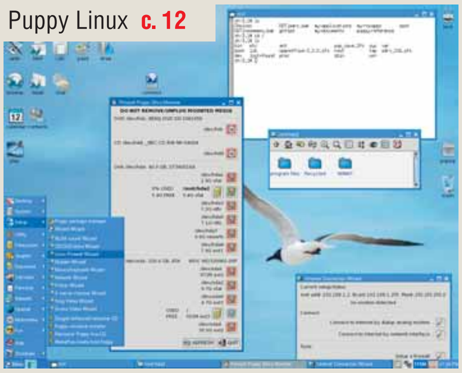
16 Storix SBAdmin 6.1
Резервирование для предприятий по цене для физических лиц? Многофункциональный, легкий в использовании и, наконец-то, стоящий своих денег пакет. Как вовремя!

Squeezebox 3 с. 15



› **Блестящая штука! Особенно этот флюоресцентный дисплей...**

Puppy Linux с. 12



НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

LINUX FORMAT TOP STUFF

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряющие практичная программа.*

Рейтинг 9/10

Ubuntu 7.04

С детства горюя о Бемби, обретет ли **Энди Хадсон** силы в Feisty Fawn? *Linux Format* попросил его установить и тщательно рассмотреть бета-версию.

Вкратце...

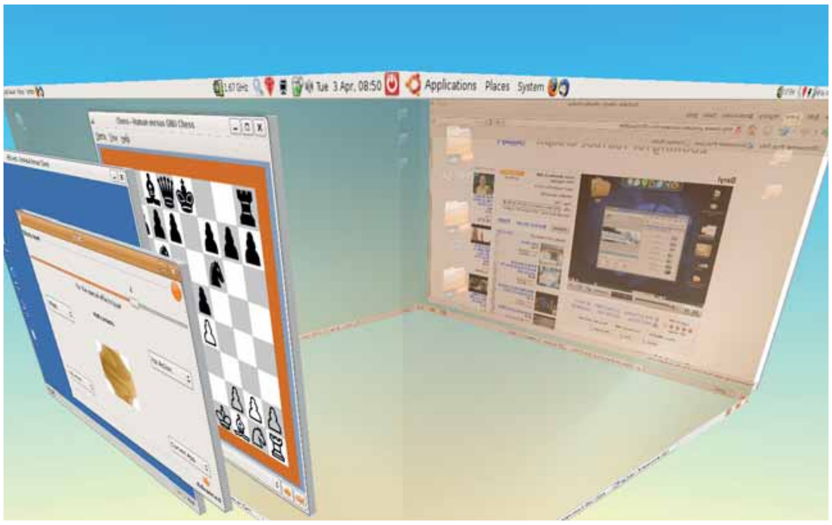
» Простой в установке и использовании Linux-дистрибутив. См. также: Mandriva, Freespire или SimplyMepis.

Мы всегда считали Ubuntu высококачественным дистрибутивом и недоумевали: ну что еще там можно улучшить? Будет ли новый продукт похож на прежние, или налицо идеология «совершенство – в переменах»?

Фактически, «Feisty Fawn [Храбрый Олененок]» – шестая инкарнация ОС: Ubuntu вышел на Linux-сцену три с половиной года назад. С самого начала Ubuntu позиционировался как «Linux для людей» и получил мощнейшую поддержку сообщества. Мы в восхищении следили за взрослением дистрибутива, особенно в предвидении жесткой конкуренции со стороны Fedora 7 и OpenSUSE 10.3. К моменту написания статьи Ubuntu 7.04 beta только что вышел, и мы не без интереса загрузили ISO и тут же вставили свежезаписанный диск в наш тестовый компьютер.

Разработчики Ubuntu подстегнули Linux-сообщество, решив выполнить свой единственный установочный диск в форме Live CD. Интересующиеся получили возможность загрузить и опробовать полноценный рабочий стол Ubuntu даже без инсталляции. Демонстрируя Ubuntu Live CD «не-Linux-пользователям» при всяком удобном случае, мы убедились, что такой подход действительно результативен, так как дает новичку возможность увидеть все собственными глазами. Версия 7.04 сохранила это качество.

> «Ubuntu 7.04 грузится за рекордное время, намного обгоняя других.»



› Ищете изюминки? Они есть у Feisty, но *Compiz* и Beryl пока не в основном составе.

Стоит отметить, что Feisty содержит проприетарные драйверы, помогающие шире раскрыть потенциал Ubuntu. В нашем случае Feisty корректно идентифицировал и настроил встроенную беспроводную карту (Intel IPW3945), а также определил беспроводную точку доступа. Об использовании закрытых драйверов уведомило полезное сообщение D-BUS. Щелчок на значке открывает (новый для Ubuntu) *Restricted Drivers Manager* (Менеджер закрытых драйверов), предлагающий выбор между установкой и удалением драйверов. Мы бы поклялись, что большинству Linux-новичков все равно, проприетарный драйвер или свободный, но, по крайней мере, Ubuntu дает возможность пользователю выбрать, а не оставляет один на один с извилистым процессом настройки.

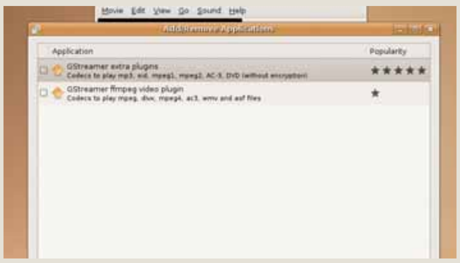
Сохранил Feisty и возможность установки с Live CD, так что среду Live CD можно легко сменить на настоящую инсталляцию. Тот, кто пользовался Ubuntu и раньше, почувствует себя здесь, как дома. Приятное дополнение, которое мы не смогли протестировать ввиду отсутствия Windows-разделов – это

Шаг за шагом: Загрузка кодеков с CodecBuddy



» Выберите файл

При попытке открыть медиа-файл, кодека для которого нет в наличии, *CodecBuddy* выдает диалоговое окно для загрузки необходимого кодека.



» Выберите кодек

CodecBuddy покажет список кодеков: просто выберите нужный. До этого, чтобы найти подходящий кодек, необходимо было пользоваться неподдерживаемым репозиторием.



» Сначала подтвердите

Перед загрузкой выдается предупреждение о том, что неплохо бы уточнить законность использования данного кодека.

мастер переноса пользовательских настроек, который якобы может перенести Windows-настройки в учетную запись Ubuntu. Это бонус для тех, кто решил перейти на Ubuntu, но не хотел бы терять персональные настройки Windows, например, избранные страницы *Internet Explorer*. Остальная часть Live-инсталляции осталась прежней, и мы установили систему на нашу тестовую машину – ноутбук Dell D620 с Core Duo (T2500, 2 ГГц), 1.25 ГГц ОЗУ и 40-ГБ жестким диском – меньше чем за 15 минут.

Энергично

Запуская свежеустановленную систему, мы первым деломы отметили быстроту загрузки. Дистрибутив грузится за рекордное время, намного обгоняя преследователей. На нашем компьютере загрузка занимает в среднем 32 секунды от нажатия кнопки до входа в Gnome (да-да, вместе со входом!). Для пользователя настольного ПК это сущий подарок, так как большинство дистрибутивов Linux отстает более чем на минуту.

Ускорение произошло, естественно, благодаря замене устаревшей *sysvinit* на *upstart*. Upstart загружает службы динамически, при необходимости, а не в порядке очереди. Хотя upstart продолжает развиваться, она уже сейчас заметно сократила время загрузки. Справедливости ради заметим, что используя Ubuntu в серверном режиме, вы не станете раздражаться из-за лишней пары минут на запуск!

Внешние изменения заметны уже начиная со входного экрана *GDM*, перекрашенного в чуть более золотистый цвет, который сказался на всей главной теме Ubuntu. Слегка. Нравится оно вам или нет, но Ubuntu не собирается снимать коричневое, хотя тему и фон можно без труда сменить за полминуты.

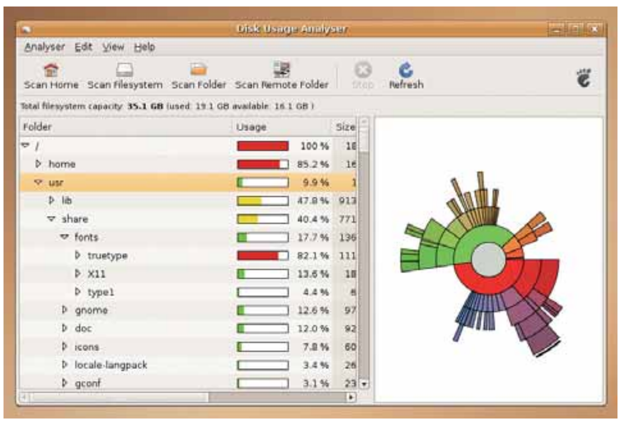
Не хватает, может быть, лишь одного – присутствия *Compiz/Beryl*, хотя взгляд в меню System > Preferences (Система > Параметры) обнаруживает наличие пункта Desktop Effects (Эффекты рабочего стола); нечто подобное применено в Fedora 6. (Кстати о меню Preferences: отметим, что в Ubuntu оно длинновато. Команда Fedora разбила его на логические группы, чтобы не метаться из конца в конец длинного списка).

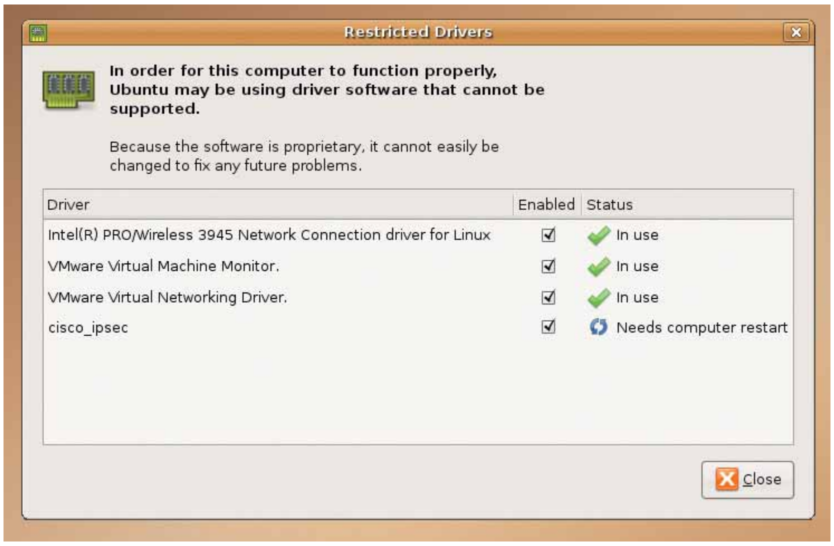


Пол считает...

«Релиз для Ubuntu скучный, и позволяет Fedora с Xen вырваться вперед. Стыдно!»

› Яркие цвета дают наглядную картину заполнения жесткого диска: заглянув вглубь файловой системы, вы узнаете, кто сожрал все доступное место!





› Спорно, зато удобно. Теперь в Ubuntu есть менеджер закрытых драйверов, облегчающий настройку оборудования.

Вернемся к *Compiz* и *Beryl*. Ubuntu до сих пор не поддерживает их официально, хотя *Compiz* установлен по умолчанию. В почти полном противоречии со своим кодовым названием, Feisty Fawn излишне робок в случае с *Compiz*, настаивая на том, что это экспериментальная функция и поэтому доступна лишь в режиме технологического анонса. Это явный перепев Red Hat Enterprise Linux 5 (*см.* **LXF92**), странный для дистрибутива без LTS (Long Term Support, долговременная поддержка). *Beryl* задвинули еще дальше, так как он доступен лишь через репозиторий Universe и не поддерживается Ubuntu официально. Правда, для желающих установить *Beryl* репозиторий Universe всяко не хуже, чем раскопки в Сети.

Досадно

Еще одна область исследования – Gnome 2.18 с полной свитой приложений. Ubuntu содержит *OpenOffice.org 2.2*, к моменту написания только что опубликованный, что дает дистрибутиву технологическое преимущество. Для фанатов виртуализации сообщаем, что Ubuntu не включает в официальный дистрибутив ни Xen, ни KVM, хотя и они доступны через репозиторий Universe.

Нас преследует ощущение некоторого разочарования в этой версии. Да, в Ubuntu многое усовершенствовано, включая *CodecBuddy* (см. *«Шаг за шагом»*, слева) и настройку беспроводных подключений через менеджер сети NetworkManager, но мы все же ожидали более яркой версии между Dapper Drake и следующей LTS. Кажется, Ubuntu-сообщество решило на сей раз не рисковать, и это позор. Список новых функций совсем куцый и пропускает вперед другие дистрибутивы – хотя бы Fedora, семимильными шагами движущуюся к выпуску седьмой версии, намеченному на конец мая. Вероятно, Ubuntu нуждается в стабильности из-за возросшего числа дистрибутивов на его основе.

Что бы ни было тому причиной, но Ubuntu теперь стоит на перепутье: мощный импульс первой волны версий (с 4.10 по 6.06) начал сходить на нет, и нужно снова как-то взбодрить этот процесс. В противном случае Ubuntu затрут новые многообещающие дистрибутивы. Куда же двигаться проекту? Одна из идей – дополнить или заменить несколько инструментов настройки Gnome, которым столь предан Ubuntu. Если разработчики действительно хотят сделать Linux удобнее, у них есть прекрасная возможность собрать комплект средств настройки «а-ля Fedora system-config-*» для базовой настройки системы пользователем.

Поймите нас правильно: мы не сомневаемся, что 7.04 будет прекрасной версией. Просто нам кажется, что Ubuntu не захотел пойти на необходимый риск, понадеявшись на свой авторитет в сообществе. LXF

LINUX FORMAT Вердикт

Ubuntu 7.04 beta

Разработчик: Ubuntu Foundation
Сайт: www.ubuntu.com
Цена: Свободно под GPL

» Версия добротная, но консервативная. Если у вас нет проблем с проприетарными драйверами, лучше остаться на 6.10.

Puppy Linux 2.16

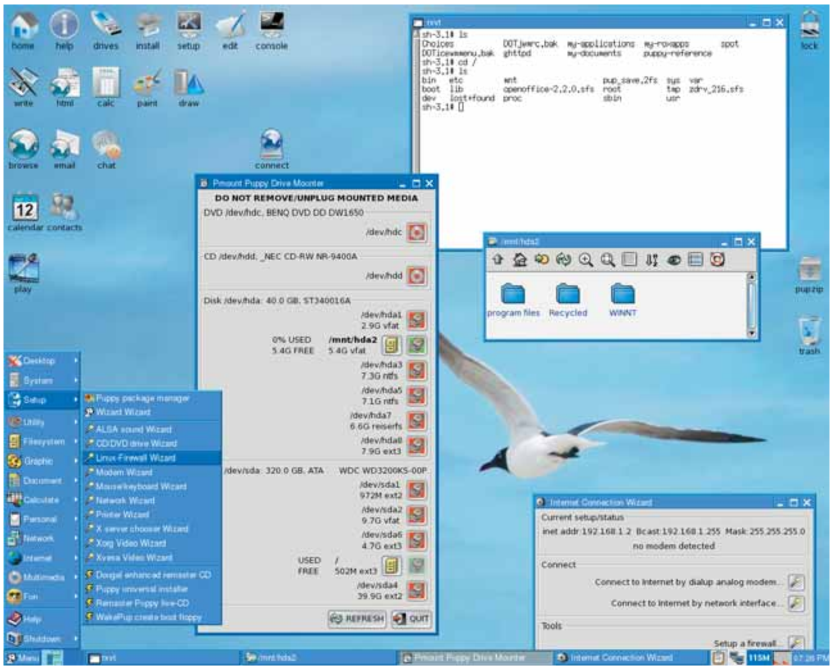
На фоне сотен дистрибутивов самого различного толка, цвета и вероисповедания почти неразличим один маленький, уникальный проект, не претендующий на мировое господство, но уверенными шажками идущий к какой-то своей цели. Однако **Сергей Супрунов** заметил его и засунул под микроскоп, чтобы рассмотреть получше...

Вкратце...

» Linux-дистрибутив весовой категории «до 100 МБ», в котором реализован ряд интересных, а порой и уникальных, идей. Помимо жесткого диска, способен работать с CD/DVD/ZIP-дисков и Flash-носителей размером от 128 МБ. См. также *Damn Small Linux*, *DeLi Linux*.

Еще одним дистрибутивом Linux сегодня вряд ли кого-то удивишь. В мире Open Source эта безграничная свобода, которую сообщество само же себе и обеспечивает, воспринимается как само собой разумеющееся. Поэтому выбирать всегда есть из чего – от миниатюрных систем, загружающихся с одной-единственной дискетки и при этом способных делать что-то полезное, до дедушки Debian, не вмещающегося и на 20 CD; от предельно дружественных Ubuntu и Mandriva до требующих определенных базовых знаний Gentoo и LFS; от следующего за всеми последними новинками Sabayon до консервативного *Slackware*...

Однако всегда находятся люди, которым недостаточно взять что-то готовое и просто пользоваться этим. И тогда они изобретают очередной велосипед. Но кто знает, может быть именно этот велосипед идеально подойдет кому-то еще? Так что попытаемся найти какие-нибудь «изюминки» в этом «еще одном дистрибутиве Linux».



› Если рабочий стол *Puppy Linux* слегка «причесать», то выглядит он очень даже мило.

Первое знакомство

Уже в меню *Isolinux* можно заметить пару нестандартных вещей (наберите какую-нибудь букву, чтобы остановить обратный отсчет автоматической загрузки и внимательно все рассмотреть). Например, указав в строке загрузчика puppy pfix=ram, можно загрузиться в режим, когда система будет полностью размещена в оперативной памяти. Это не только освободит привод для чего-нибудь более полезного, но и обеспечит потрясающую скорость работы. Тем более что небольшой размер дистрибутива практически не создаст неудобств обладателям более-менее современных компьютеров.

Еще один сюрприз будет поджидать вас после загрузки ядра: вам предложат выбрать графический сервер – *Xorg* или *Xvesa*. С первым все понятно, он уже давно стал фактическим стандартом любого Linux-дистрибутива с графическим окружением. А вот уже экзотический Vesa-сервер может оказаться полезным там, где ресурсы компьютера не позволяют полноценно использовать «тяжеловеса» *Xorg*.

Нужно отметить и один неприятный момент – при выборе *Xorg*-сервера Puppy далеко не всегда справляется с автоматическим определением оптимального режима, и приходится выбирать желаемое разрешение и глубину цвета вручную (что, впрочем, не слишком обременительно, если знать параметры своего оборудования).

Среда JWM (Joe's Window Manager) загружается довольно быстро – на машине AMD64 3000+ уже через 15 секунд система встретит вас приветственным лаем, оповещающим, что звуковая карта определилась и готова к работе. Трехмерные «стеклянные» элементы управления выглядят несколько не к месту среди архаичного *GTK*-интерфейса, хотя и несколько разнообразят окружающую «серость». Впрочем, достаточно выбрать в JWM Configuration Manager тему Puppy-XP, и все становится совсем замечательно.

Почему-то в дистрибутивах, предлагающих в качестве рабочего окружения ту или иную «легковесную» оболочку, обычно сразу настраиваешься на большой объем ручной работы – кажется, что подобные проекты делаются исключительно хакерами для хакеров. Но в *Puppy Linux* все ориентировано на вполне обычного пользователя – целая россыпь различных «мастеров» помогает настроить мышкой почти все, что нужно для повседневной работы: раскладки клавиатуры, внешний вид рабочего стола, звук, принтеры, сетевые подключения, даже брандмауэр. Правда, нужно заметить, что некоторые «мастера» работают в текстовом терминале, предоставляя пользователю псевдографический интерфейс. Впрочем, это нисколько не мешает решению основной задачи. Столь же легко система позволяет работать и с пакетами дополнительного ПО.

Искусство упаковки

В *Puppy* реализован подход к управлению пакетами, во многом напоминающий PBI, используемый в PCBSD – вы выбираете в менеджере пакетов нужные, щелкаете по кнопке OKAY, указываете, откуда брать **pet**-файлы (к

Ключевое ПО Puppy

(включенное в основной образ)

Ядро 2.6.18.1	Abiword 2.4.5
Xorg 7.0.0	Gnumeric 1.6.3
JWM v1.8	Seamonkey 1.0.8
ROX-Filer 2.5	gxine 0.5.9
BusyBox v1.01	

вашим услугам несколько ftp-серверов, а также возможность использовать локальное хранилище), при необходимости ждете несколько секунд/минут, пока завершится загрузка выбранных пакетов, еще несколько секунд на инсталляцию – и все готово. Никаких сборок, зависимостей и т.д. Возможно, не столь гибко, как хотелось бы опытным пользователям, зато быстро и без лишних вопросов.

В более ранних версиях использовались **pup**-пакеты (менеджер DotPup), и кое-где в инструментарии и документации вы можете встретить упоминание о них. Однако на ftp-сервере их не осталось, и к использованию они не рекомендуются.

Но и это еще не все – вы можете использовать **sfs**-образы (SquashFS) для расширения возможностей дистрибутива. Например, таким способом устанавливается *OpenOffice.org* – просто поместите **sfs**-файл в каталог **/mnt/home**, отметьте его в BootManager (пункт меню System), и при следующей загрузке вы обнаружите соответствующий пакет готовым к работе.

Ничто не забыто

Ну вот – потрачено несколько драгоценных минут на то, чтобы привести систему в соответствие со своими потребностями и пожеланиями. А теперь вспомним, что мы работаем в режиме LiveCD... Неужели при следующей загрузке все придется начинать заново?

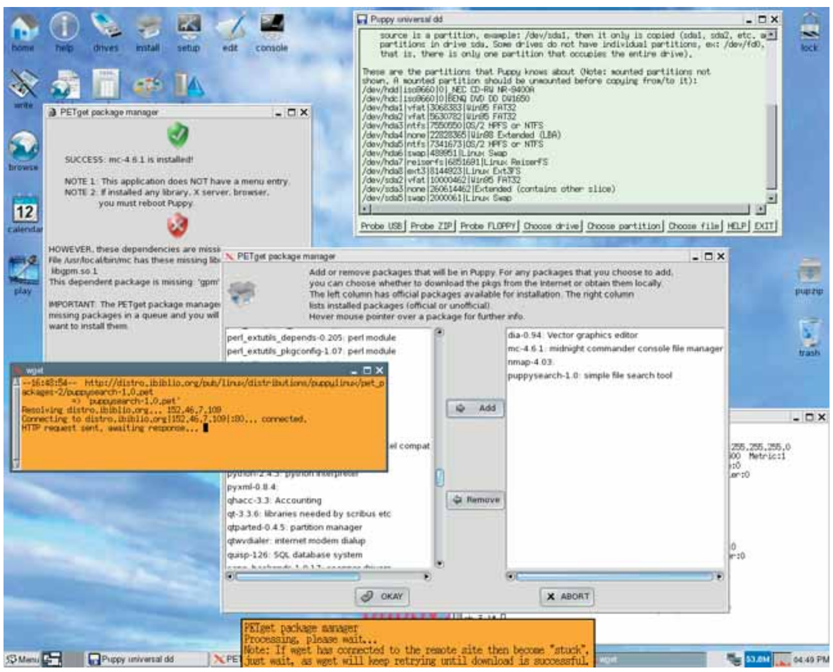
Разработчики *Puppy Linux* приготовили целых три пути для решения этой проблемы. Это, во-первых, традиционное сохранение файла настроек на диск, «флешку», CD-R, в общем, где найдете место. При желании сохраняемую информацию можно зашифровать. Впрочем, это умеют почти все LiveCD, и ничего удивительного здесь нет. Но *Puppy Linux* не может не удивлять, и этот случай – не исключение. Поэтому...

Во-вторых, вы можете сохранить свои настройки прямо на тот же диск, с которого работаете! Да-да, ISO-образ Puppy записывается в режиме multisession, так что вы можете просто дописывать все, что нужно, ему в «хвост» – действительно, не пропадать же целым 500 МБ дискового пространства? Понятно, что для этого вам нужен пишущий CD-привод. При следующей загрузке вы сможете дополнительно указать, сколько сессий следует пропустить перед поиском сохраненных файлов.

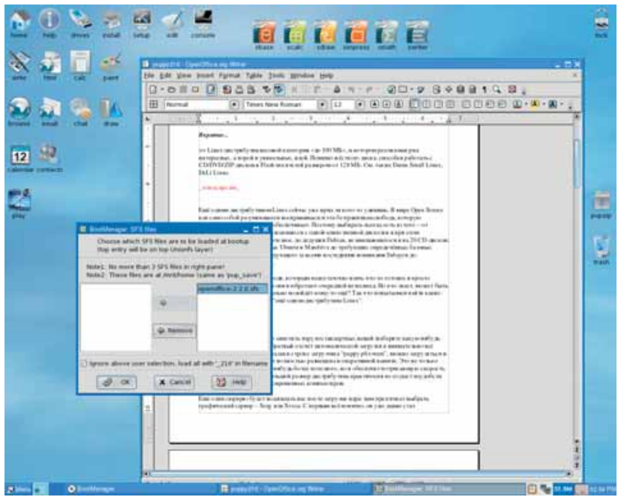
Но есть еще и «в-третьих»! Выбрав пункт Remaster CD в разделе Setup, вы сможете переписать образ дистрибутива, по новой сохранив его со всеми вашими настройками, установленными дополнительно программами и т.д. То есть вы, фактически, получаете возможность создать свой собственный дистрибутив, совершенно не задумываясь о таких вещах, как базовая система, загрузчики, файлы initrd и прочее. Просто настройте систему, как хочется, и запишите результат на CD/DVD-диск или в ISO-образ.

Мирное сосуществование

Кстати, если вы решите обеспечить для *Puppy Linux* постоянную прописку на своем винчестере, то и это делается элементарно – запустите Puppy universal installer из подменю Setup и следуйте несложной процедуре. Проницательный читатель уже, наверное, догадывается, что и здесь где-то должен быть сюрприз. И он действительно есть – помимо традиционной установки на отдельный раздел (в инсталляторе этот режим называется NORMAL), можно выбрать режим COEXIST, в котором *Puppy Linux* будет установлен на раздел с существующей ОС (причем это может быть даже Windows XP на разделе NTFS; правда, придется с загрузчиком слегка повозиться), оставив последнюю в целости и сохранности. Puppy просто размещает на указанном разделе нужные для работы файлы-образы, которые размещаются в памяти при загрузке системы, подобно тому, как это происходит при работе с CD-диска.



› Вот так происходит инсталляция нового ПО – пришел, увидел, установил.



› Укажите sfs-образы, и при следующей загрузке все будет на своих местах. Кстати, информация к сведению – *OOo Writer* загружается 2 (две) секунды!

Итак, *Puppy Linux* оказался вполне пригодным для повседневной работы. Множество интересных решений не только упрощает такие задачи как установка различных программ и работа с LiveCD с сохранением текущих настроек и наработок при переходе от одной машины к другой. Они также дают уйму идей, которые, возможно, кто-то возьмет на вооружение в своих проектах.

Прибавьте к этому легковесность, молниеносную загрузку, отсутствие лишнего и наличие нужного... В общем, если иногда возникает желание быстро заглянуть в Интернет, что-то подредактировать или проверить электронную почту, не дожидаясь, пока загрузится «основной» дистрибутив, то установка *Puppy Linux* в режиме COEXIST – почти идеальное решение. LXF

LINUX FORMAT Вердикт

Puppy Linux

Разработчик: Барри Каулер и сообщество
Сайты: www.puppyos.com; www.puppylinux.org
Цена: бесплатно под GPL

Функциональность	7/10
Производительность	10/10
Простота использования	9/10
Документация	8/10

» *Превосходен на старых компьютерах, молниеносен на современных, удобен при частых перемещениях. Так что подумайте, а зачем вам SUSE?*

Рейтинг 9/10

Scribus 1.3.3.8

Scribus – лучшая издательская система для Linux, больше того: это единственная издательская система для Linux. Но может ли она соперничать с великими? Разбирается **Ник Вейч**.

Вкратце...

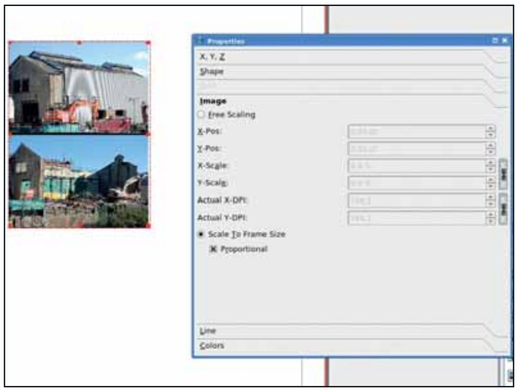
» *Scribus* как программа профессиональной верстки уникальна в своем роде. Для документов попроще есть *OpenOffice.org*, а для суперсложных – *TeX*.

Мы описывали *Scribus* в разделе HotPicks LXF36. Тогда мы расценили его как интересный, амбициозный и многообещающий проект. Теперь смело скажем: мы не ошиблись. Казалось невероятным, что на DTP-Олимпе можно потеснить таких гигантов, как *QuarkXPress* и *Adobe InDesign*, но после ряда версий функциональность *Scribus* настолько развилась, что программа стала реальной альтернативой для многих настольных систем. *Scribus*, инсталлированный нами, имеет расширенную поддержку графики, управление цветом, переработанный GUI, поддержку SVG, генератор штрих-кодов и многое другое. Поддержка PDF великолепна, с настройками 1.3/1.4/1.5, шифрованием и даже презентационными эффектами.

Одно из главных достижений команды *Scribus* – одновременная разработка вариантов программы для трех ведущих настольных систем: Linux, Windows и Mac. Помимо возможности выбора платформы (ну, мы-то с вами знаем, какая лучше), такой метод разработки позволяет предоставить программу гораздо большему числу пользователей самых разных уровней подготовки, чем в случае с обычным приложением Linux. Система Mac была неотъемлемой частью процесса DTP с момента возникновения самой его идеи. Поколения искусных верстальщиков выросли, не зная ничего, кроме MacOS, поэтому развитие именно в этом направлении представляется наиболее важным.

Охота за ошибками

Это не окончательная стабильная версия, и критиковать разработчиков было бы несправедливо. Да, здесь полно недоработок (которые непрестанно исправляются), но это скорее мелкие неприятности, чем крупные провалы. Например, откат действия Adjust Frame To Image (Подогнать рамку по рисунку) на самом деле не приводит к изменению размера рамки. Но нет сомнений, что все эти мелочи исправляются сразу после обнаружения.

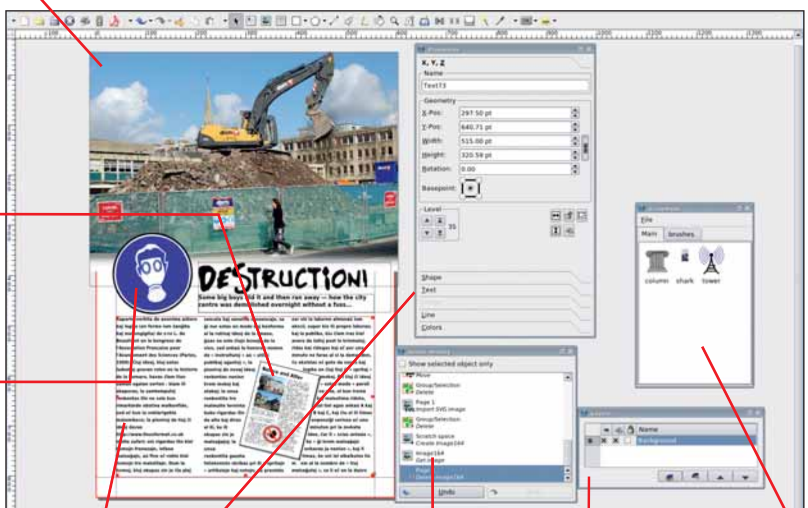


› Для настройки изображения придется не раз прогуляться в окно Properties.

Более глубокую озабоченность вызывает пользовательский интерфейс. *Scribus* нуждается в тщательном эргономическом исследовании. Все основные функции DTP имеются, да только найти их не всегда легко. Возьмем, к примеру, текст. Настройка количества колонок в текстовом поле находится в разделе Shape (Форма) диалогового окна Properties (Свойства), а это для нее не самое очевидное место. В том же диалоговом окне можно настроить толщину и стиль контурной линии, но отображаться она не будет, так как цвет по умолчанию – белый, а цвета настраиваются в другой вкладке.

Scribus отнюдь не неудобен в работе, но чтобы завоевать признание придирчивых верстальщиков, необходимо глубоко и вдумчиво проработать интерфейс. LXF

Исследуем интерфейс Scribus



Импорт изображений *Scribus* поддерживает общие стандарты растровой графики, включая профили управления цветом и прочее.

Поддержка PDF Наряду с экспортом PDF-документов, *Scribus* может еще и читать их!

Модули Модули обеспечивают дополнительные функции, импорта SVG.

Слова У *Scribus* развитые типографские возможности и удобный редактор Story Editor.

Properties (Свойства) Основная часть манипуляций с объектами происходит здесь.

History (Журнал) Диалоговое окно Журнал – окно в прошлое документа.

Слои Упрощают управление сложными документами.

Сохранение В этом буфере хранятся элементы, которые часто требуются в работе.

Вердикт

Scribus 1.3.3.8

Разработчик: The Scribus team
Сайт: www.scribus.net
Цена: Бесплатно под GPL

Функциональность	8/10
Производительность	8/10
Простота использования	6/10
Документация	6/10

» *Потенциально велик, но нуждается в доводке интерфейса.*

Рейтинг 8/10

Squeezebox 3

Проблема прослушивания музыки на PC всего одна – бежевый Linux-компьютер может не сочетаться с вашими занавесками. Решение ищет **Грэм Моррисон**.

Вкратце...

» Передавайте музыку со своего Linux-сервера на блестящий плейер. Возможные альтернативы: *MythTV* или обновленная версия Hauppauge MediaMVP.

Реальный хакер может думать о Squeezebox как о тонком клиенте потокового вещания. Для остального человечества это сетевой музыкальный плейер, подключаемый к hi-fi аппаратуре и воспроизводящий музыку прямо с сервера. Первое поколение плейера увидело свет в 2001. В то время к сети он подключался по проводам, а внешностью напоминал переносную армейскую рацию.

Сейчас перед вами третье поколение, и в нем внешний лоск сочетается с техническими новинками: беспроводным подключением 802.11g и завораживающим вакуумным флуоресцентным дисплеем 320x32. Оживив дисплей включением питания, оторваться от него уже невозможно.

Но лучшая черта Squeezebox и причина его появления в нашем журнале – программная часть этого плейера работает под Linux. Она имеет открытый исходный код, часто обновляется и имеет больше функций, чем пикселей на дисплее. Ну, почти.



› **Squeezebox третьего поколения не выглядит белой вороной рядом с hi-fi системой.**

От Hi-Fi до Wi-Fi

ПО называется *SlimServer* и поставляется в виде RPM или Deb-пакетов. Новейшая версия – 6.5, седьмая проходит бета-тестирование. Единственная зависимость – Perl 5.8.3 или выше. Инсталлированный сервер доступен через web-интерфейс, подключенный к порту 9000. Первым делом нужно указать серверу, где находится музыкальная коллекция, с помощью панели управления основными настройками (Basic Settings). *SlimServer* воспроизводит файлы MP3, Ogg Vorbis, Flac и WAV. Сканирование потребует некоторого времени, в зависимости от размеров коллекции: *SlimServer* будет составлять базу данных из всех тэгов, встроенных в музыкальные файлы, чтобы рассортировать коллекцию по исполнителям, альбомам и жанрам.

Сила *SlimServer* в том, что его можно использовать без аппаратуры. Подключите любой браузер к '9000/stream.mp3', и слушайте свою музыку, где угодно. Да, хакеры, он работает через SSH-туннель; есть и страница для узких каналов, используемая с PDA. Есть даже встраиваемый программный клиент под названием *SoftSqueeze*, создающий иллюзию, что Squeezebox действительно стоит у вас на столе. Это практически идентичное воспроизводство, вплоть до пульта ДУ.

Музыка для вас

Настроив сервер, остается лишь сообщить устройству о том, где он находится – и готово. Чтобы все заработало, совсем не нужно быть экспертом. Можно подумать, что однолинейный интерфейс Squeezebox ограничен, но это не так. Следуйте иерархии меню, переходя между параметрами курсором вверх и вниз и выбирая параметр курсором вправо; а курсор влево осуществляет возврат на предыдущий пункт. В согласии с вашими действиями экран плавно прокручивается. Например, в главном меню можно выбрать критерии отбора файлов: по исполнителям, по альбомам, по годам и по папкам, в которых хранится музыка. В любое время можно нажать Play, и Squeezebox добавит все, что ниже курсора, к списку воспроизведения.

Лучшее свойство навигации – отзывчивость: вспомните, как нудно просматривать коллекцию, например, на ноутбуке. Вот такие мелочи и определяют впечатление. Выключите устройство, затем нажмите Play – прибор включится и начнет воспроизведение немедленно. Предположительно, Squeezebox использует 64 МБ памяти для кэширования музыкальных данных. Можно слушать потоковое интернет-радио из тысяч источников, играть в *Тетрис* и читать новости с RSS. Но главное – музыка звучит фантастически. LXF

Задняя панель



LINUX FORMAT Вердикт

Squeezebox 3

Разработчик: Slim Devices
Сайт: www.slimdevices.com
Цена: около $400

Функциональность	10/10
Производительность	9/10
Простота использования	7/10
Цена	8/10

» *Дорого, к тому же нуждается в Linux-сервере, но для меломана – просто мечта.*

Рейтинг 9/10

Storix SBAdmin 6.1

В вечной погоне за хорошими инструментами администратора **Грэм Моррисон** нашел, что серьезный пакет для резервного копирования не обязан стоить серьезных денег.

Вкратце...

» Программа резервного копирования для обычных пользователей и системных администраторов; допускает удаленное управление и справляется с необработанными данными. См. также: *BakBone NetVault* или *BrightStor ARCserve*.

Storix System Backup Administrator, или *SBAdmin* – полноценный комплект для резервного копирования, созданный, чтобы всемерно упростить эту нелегкую работу. По первому впечатлению, инсталляция не настолько проста, как хотелось бы. Для вызова оболочки по умолчанию инсталляционный скрипт использует */bin/sh*, тогда как синтаксис скрипта в действительности требует Bash. Конечно, большинство дистрибутивов по умолчанию использует Bash, но ведь не все. Например, в Ubuntu */bin/sh* – это символическая ссылка на Dash, выбранную по соображениям экономии ресурсов, и перед запуском скрипта ее нужно перенаправить на Bash.

Вторая – и последняя – проблема состояла в скрытой зависимости от пакета **libpcre3**, используемого при инсталляции SSL-сертификата для web-интерфейса. После установки пакета все остальное работало как полагается, и мы нашли программу в целом очень стабильной – что и требуется от подобного приложения.



› **В отличие от некоторых, пользовательский интерфейс здесь прост.**

Всего за два щелчка

Общаться с программой можно тремя способами. Можно использовать один из доброй дюжины инструментов командной строки и построить собственную стратегию резервирования, но в случае с *SBAdmin* это будет излишним. В отличие от аналогичных Linux-программ (*NetVault* от *BakBone*, или *BrightStor ARCserve Backup*), Storix попыталась построить свое ПО с учетом навыков среднего Linux-пользователя. Это подразумевает простой графический пользовательский интерфейс, на наш взгляд – один из самых удобных (только вот шрифты пришлось увеличить).

Основной процесс включает в себя создание задания на резервирование: определение источника и папки назначения, где вы собираетесь хранить результаты; назначение заданию автоматически генерируемого ID; назначение расписания. Можно выбирать из нескольких уровней сжатия, но даже при вышем уровне оказалось, что наши пользовательские данные утрамбовались лишь на 20% – bzip2 или gzip сработали бы лучше. Преимущество такого подхода – быстрота: наш домашний раздел объемом около 5 ГБ был упакован за несколько минут, а задания сохранены и готовы к перезапуску.

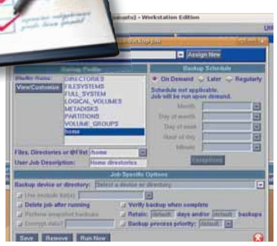
После добавления задания в очередь оно появляется в главном окне состояния, и на этом все действительно заканчивается. Что особенно впечатляет – основной GUI был воссоздан на встроенном web-сервере, поэтому получить доступ к очереди заданий можно отовсюду с помощью браузера и сетевого подключения к серверу.

За внешней простотой скрывается серьезная сила. В противоположность инструментам резервирования KDE и Gnome, *SBAdmin* богат функциями. Поражает способность упаковывать и восстанавливать не только папки и файлы в файловой системе, но и необработанные данные на дисках и разделах. Эта возможность пригодится при резервировании сервера, когда диски в случае катастрофы могут быть восстановлены в точности такими, какими они были, включая состояние ОС. Функция миграции устройств позволяет восстановить резервную копию даже на другом оборудовании.

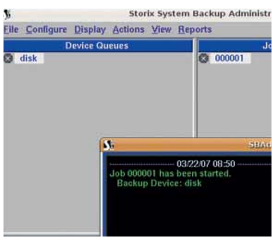
SBAdmin неплохо смотрится на фоне других средств корпоративного уровня. Как и NetVault, *SBAdmin* поддерживает автоматизированные ленточные накопители, а также резервирование и восстановление по сети. Не стоит ставить *SBAdmin* в один ряд с программами уровня предприятия, но для средней офисной сети приложение более чем пригодно. LXF



Свойства навскидку



Милое дело!
Создать задание проще простого: выберите данные, укажите папку назначения и назначьте номер задания.



Выбор интерфейса
Автоматическое расписание заданий можно установить и в главном окне программы, и через web-интерфейс.

LINUX FORMAT Вердикт

Storix SBAdmin 6.1

Разработчик: Storix Inc
Сайт: www.storix.com
Цена: Linux/x86 Workstation Edition $459 (без НДС)

Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Цена	9/10

» *Больше функций на фунт, чем у соперников; идеальная цена для малых предприятий.*

Рейтинг 9/10

Mandriva Linux

Linux center
www.linuxcenter.ru



Mandriva Corporate Server

Серверное решение Mandriva, первая версия которого была выпущена еще в 2000 году. Являясь частью корпоративной линейки, Mandriva Corporate Server имеет удлиненный жизненный цикл (12-18 месяцев) и поддерживается производителем в течение пяти лет. Благодаря использованию фирменных мастеров Mandriva система очень проста в обслуживании, а автоматическая установка по сети (PXE Auto Install) делает ее развертывание более быстрым.



Mandriva Corporate Desktop

Настольное решение в линейке корпоративных продуктов Mandriva. Как и Mandriva Corporate Server, имеет более длинный жизненный цикл (12-18 месяцев) и поддерживается производителем в течение пяти лет. Разработчики приложили все усилия, чтобы максимально упростить развертывание системы на предприятиях: инсталлятор Mandriva Corporate Desktop может работать в автономном режиме, кроме того, возможна автоматическая установка по сети (режим "PXE Auto Installation").

Дистрибутив Mandriva (ранее, до слияния с Connectiva называвшийся Mandrake), по праву считается идеальным для неподготовленного пользователя. Mandriva Linux невероятно прост в установке благодаря современному графическому установщику. Гордость дистрибутива – комплект графических утилит для настройки системы, объединенные в Mandriva Control Center. Установив Mandriva, вы сможете настроить абсолютно все, от звуковой карты до файерволла, не прибегая к помощи командной строки. Разработчики Mandriva Linux очень тщательно подходят к тестированию программных пакетов, поэтому в новые версии дистрибутива выходят сравнительно редко, но зато содержат только стабильные программы.

Сравнение

Каждый месяц мы сравниваем тонны ПО – а вы можете отдыхать!

Командные оболочки

Не забирались дальше *Bash*? Оболочка – не просто поле для ввода команд, так что познакомьтесь с парочкой экземпляров из коллекции **Нейла Ботвика**.



Про наш тест...

Оболочки используются двумя способами. При интерактивном применении самым важным аспектом является простота: предоставляется ли стандартное или расширенное дополнение, хороша ли документация, есть ли подсказка? Другой способ использования – в качестве движка для запуска сценариев оболочки. Здесь основными требованиями будут, во-первых, совместимость со стандартами, а во-вторых – и это менее важное требование – скорость.

Для оболочек не существует стандартных тестов, да если бы они и были, оболочки вроде *Bash* и *Zsh* имеют более быстрые альтернативы функций POSIX, поэтому справедливо ли их сравнивать? Мы запустили тестовый скрипт с несколькими циклами, содержащими парочку логических тестов, чтобы определить относительную скорость. Поскольку некоторые оболочки используют нестандартный синтаксис скриптов, ранжировать их не удалось, но там, где скорость достойна упоминания, она отмечена в индивидуальных обзорах.

Тестирование производилось на ПК Athlon64 3200 с 1 ГБ ОЗУ, и на 1-ГГц G4 iBook с 768MB, на обоих был установлен Gentoo Testing.

Наш выбор

Bash с. 19
Dash с. 21
Dsh с. 22
Fish с. 20
Sash с. 22
Tcsh с. 20
Zsh с. 21

Разве мы не обсуждали терминалы совсем недавно? И есть ли разница между оболочкой и терминалом? Ответы – да, в обоих случаях. Оболочка – это программа, работающая в терминале. Она отображает приглашение для ввода команд и пытается выполнить то, что вы ей вводите. Возможно, разница не столь очевидна в X-терминале, где вы регистрируетесь автоматически, но в виртуальной консоли (в том экране, который вы видите, когда X не запущен), выводящей текстовое приглашение для входа в систему, оболочка не запускается до тех пор, пока вы не введете правильные имя пользователя и пароль.

Каждый пользователь может работать в своей оболочке, так что себе вы можете поставить «мощную» оболочку, а для своей бабули предусмотреть более понятную. Даже если вы сроду не вводили команды в окно терминала, оболочкой вы все же пользовались: многие программы на вашем компьютере фактически являются ее сценариями, так что запуская программу, вы каждый раз запускаете оболочку, даже если делаете это, нажимая на значок.

Оболочки всех размеров

Рассмотренные здесь оболочки делятся на три основных категории. Оболочки общего назначения, типа *Bash* и *Zsh*, предлагают широкий выбор функций и свойств. Оболочки-легковесы, например, *Dash*, предлагают самые основные функции в небольшом пакете, и хороши для старых ПК или систем с ограниченными ресурсами; это также хорошие кандидаты на интерпретацию скриптов, которым не нужны «навороты» *Bash* со товарищи. Третья категория, оболочки особого назначения, созданы для специальных приложений или сред. На этих страницах мы рассмотрим образчики каждой категории и сравним их пригодность для решения определенных задач.

Кроме внешних команд, каждая оболочка умеет выполнять некоторое количество собственных команд ('builtins'), ускоряющих выполнение основных операций. Большинство рассматриваемых оболочек соответствуют основному стандарту POSIX. Различаются они предлагаемыми дополнительными функциями. Это могут быть дополнительные builtin-команды, функции типа расширенного дополнения или попросту минимальная среда POSIX в компактной и быстрой программе.

Bash

Фаворит дистростроителей, но вправду ли он лучше всех?



› **Man-страница *Bash* огромна, но версия info разбита на разделы разумных размеров.**

В начале был *Bourne Shell* (*sh*), его написал Стивен Борн [Steven Bourne] для Bell Labs Research Unix. *Bash* – это *Bourne Again Shell* (Снова Оболочка Борна), ужасный каламбур, который, к счастью, редко используется. Почти все современные дистрибутивы Linux используют *Bash* в качестве оболочки по умолчанию, и вы, вероятно, знакомы как минимум с его основами; это превращает *Bash* в фактический стандарт, с которым сравниваются все остальные. Поэтому имеет смысл для начала рассмотреть *Bash*, а затем перейти к другим оболочкам, изучить, чем они отличаются, и выделить их достоинства и недостатки. Итак, что же дает нам *Bash*, и почему *Bash* столь популярен у разработчиков дистрибутивов?

Дело не в малом размере *Bash*, и не в скорости. По размеру *Bash* больше всех рассмотренных здесь оболочек, кроме одной: *Sash*, которая не использует библиотек и имеет несколько дополнительных встроенных команд. *Bash* также и не самая быстрая оболочка, которую вы встретите (если, конечно, не отшвырнете журнал, проигнорировав остаток нашего **Сравнения** и поклявшись всю оставшуюся жизнь пользоваться только *Bash*). Однако большинству пользователей на это наплевать, ибо подлинно важна его гибкость. *Bash* обладает некоторыми функциями, превосходящими стандарт POSIX, хотя при желании можно добиться от него и POSIX-поведения. Если запустить *Bash* командой sh, с опцией командной строки --posix или при установленной переменной окружения POSIXLY_CORRECT, *Bash* будет работать как стандартная оболочка POSIX. При запуске через sh, *Bash* по возможности пытается работать как исходная оболочка Борна, но лишь в тех ситуациях, когда это не вступит в конфликт со стандартом POSIX.

> «Bash обладает некоторыми функциями, превосходящими стандарт POSIX.»

Встроенные команды позволяют *Bash* выполнять некоторые операции быстрее, ценой чуть более долгого времени загрузки, но если вы запускаете скрипт *Bash* из оболочки *Bash*, об этом и говорить не стоит. Эти команды плюс дополнительные логические конструкции означают, что скрипты, написанные для *Bash*, могут работать быстро. Например, типичный способ запустить цикл с установленным количеством повторений – for i in $(seq 1 100), при этом внешняя команда seq генерирует список повторений. С помощью *Bash* вы можете написать for i in {1..100}, при этом не вызывается никакая внешняя программа и все выполняется быстрее.

О POSIX

Portable Operating System Interface или POSIX представляет собой собрание стандартов, определяющих совместимость операционных систем. POSIX-совместимая оболочка отвечает соответствующим частям стандарта и будет работать с любым скриптом, написанным в соответствии с ним. Обычно POSIX считается основой оболочки, к которой добавляются дополнительные функции. Большинство рассмотренных здесь оболочек соответствуют стандарту, но некоторые для его обеспечения требуют указания ключа в командной строке. Стандартная оболочка – это обычно **/bin/sh**, поэтому многие оболочки добавляют это в качестве символической ссылки на самих себя. Запуск оболочки через **/bin/sh** включает полную POSIX-совместимость.

RTFM

Набрав **help**, вы получите список всех встроенных команд. Добавив имя команды, вы получите информацию об ее использовании. Это устраняет необходимость лазить по man-странице *Bash*, содержащей явный переизбыток информации. Тут *Bash* может поучиться у *Zsh*, где документация распределена по нескольким страницам. Документация, представленная в формате info, удобнее для обработки, при условии, что вы умеете пользоваться info, но многим из нас это недоступно. Пользователи KDE могут легко просмотреть документацию, введя 'info:*Bash*' в *Konqueror*.

Как и большинство оболочек, *Bash* включает функцию стандартного tab-дополнения: нажмите Tab на первом слове, и он попытается дополнить его до имени команды, нажмите Tab на следующем слове, и он попробует дополнить его до имени файла или пути. Однако в плане программируемого дополнения *Bash* может пойти намного дальше. Как только вы установите и добавите в свой профиль дополнительный пакет из **www.caliban.org/Bash/index.shtml#completion**, функция tab-дополнения будет завершать переменные в соответствии с командой, за которой они следуют. Например, введите ssh, а потом несколько первых букв имени хоста в вашей сети, и *Bash* дополнит его на основе содержимого вашего файла SSH known_hosts. Программируемое дополнение также можно использовать в качестве командной опции; введите somecommand --TAB – и увидите список опций, доступных в этой программе. Дополнение работает и с неопциональными аргументами для программ. Все это, конечно, чудесно, но не забудьте, что у программ должна быть специальная поддержка: обычно это файл в **/usr/share/Bash-completion**. Некоторые из них прилагаются, а некоторые доступны онлайн.

Легко понять, почему разработчики дистрибутивов так любят *Bash*. Во внутренней работе системы многое (например, все скрипты сервисов, запускаемых при включении и выключении) требует эффективного использования сценариев оболочки, и удобно программируемый *Bash* обречен на сильную любовь – да такую сильную, что некоторым дистрибутивам эта оболочка требуется в обязательном порядке, и без нее они работать не будут. Впрочем, вы не обязаны выбирать себе *Bash* в качестве интерактивной оболочки. Конечно, в нем широкий спектр функций и налицо удобство использования, но если вы просто получили *Bash* вместе с дистрибутивом и не пробовали ничего другого – читайте дальше и, возможно, найдете что-нибудь получше!

LINUX FORMAT Вердикт

Bash

Версия: 3.2_p15

Web: http://cnswww.cns.cwru.edu/~chet/bash/bashtop.html

Лицензия: GPL

» *Отлично подходит для скриптов и хорош для всех, кроме самого требовательного интерактивного пользователя.*

Рейтинг 8/10

Fish

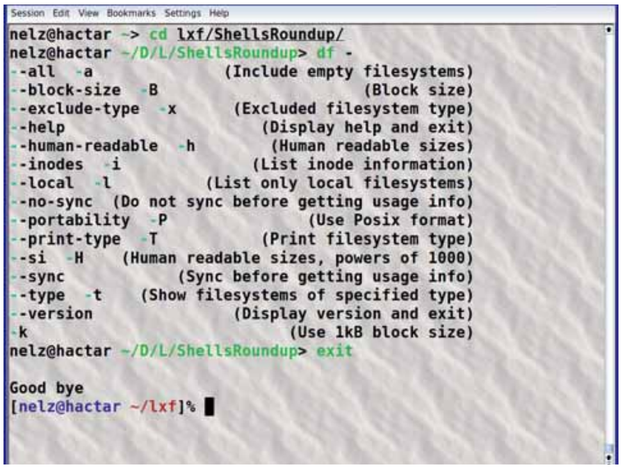
Нестандартные скрипты, но идеален для новичков.

Работа с оболочкой слегка пугает начинающих пользователей. Поэтому *Fish*, *Friendly Interactive Shell* (Дружелюбная Интерактивная Оболочка), даже предлагает привыкшим к графическому пользовательскому интерфейсу ввести help после входа. Эта команда при первом использовании попросит вас выбрать браузер, а потом будет открывать в нем документацию в формате HTML. Подробная и доступная информация о командах делает их поиск значительно проще, чем в man-страницах оболочек, подобных *Bash* или *Zsh*. Если *Fish* не сможет подключиться к X-серверу, для показа справки будет задействован links.

Документация – не единственная дружелюбная функция в *Fish*. Здесь также используется подсветка синтаксиса: когда вы начинаете вводить команду, буквы показаны красным, и меняют цвет на зеленый при правильном завершении имени команды, а имена действующих директорий подчеркиваются. По ощущениям (без измерений), Tab-дополнение вроде бы быстрее, чем в *Bash*, и почти не уступает программируемому дополнению *Bash*, но не столь расширяемо; оно работает над командными опциями, так же, как и над самими командами. Предыстория команд тоже более интуитивна: просто введите начало команды и нажмите стрелку «вверх», чтобы просмотреть предыдущие вызовы – это проще, чем запомнить Ctrl-R и иже с ним.

Fish использует специфические конструкции скриптов, которые проще для понимания начинающих пользователей, но несовместимы с другими оболочками. Это не фатальный недостаток, потому что *Fish* предназначен для интерактивного использования, хотя эти конструкции также могут использоваться в командной строке. В защиту *Fish* можно сказать, что он ориентируется на тех, кого вряд ли интересуют подобные функции, и достаточно гуманен, чтобы подробно объяснить, почему первая форма не работает, и привести пример корректного (для *Fish*) формата. Да, *Fish* избавился от некоторых загадочных приемов в стиле Unix, но вам все же придется держать под рукой *Bash*, чтобы запускать системные скрипты.

> «Подсветка синтаксиса весьма удобна для привыкших к графике.»



› *Fish* и дружелюбен к новичкам, и вежлив – он даже мило прощается с вами при выходе.

LINUX FORMAT Вердикт

Fish
Версия: 1.22.2
Web: http://fishshell.org
Лицензия: GPL

» *Очень дружественна: даже ваша бабушка справится с нею. Хороший первый выбор для начинающих пользователей оболочек.*

Рейтинг 8/10

Tcsh

Оболочка C Shell. Приглядитесь!

На первый взгляд, *Tcsh* весьма похож на *Bash* и *Zsh*, ибо включает дополнение команд и имен файлов, предысторию команд с функцией поиска и проверку написания – но для своих скриптов он использует совершенно другой синтаксис. Поэтому, хотя *Tcsh* и взаимозаменяем с другими оболочками по части интерактивного использования, вам все же понадобится *Bash* для запуска скриптов, не написанных специально для *Tcsh*.

Как и *Bash* с *Zsh*, *Tcsh* использует по умолчанию привязку клавиш в стиле *Emacs*, но при желании ее можно переключить на vi (фанаты *vi*, не принимайте, пожалуйста, это за восхваление *Emacs*, и не развязывайте очередную религиозную войну редакторов). *Tcsh* не предлагает программируемого или опционального дополнения, как *Zsh* или *Bash*, однако у него есть функция проверки написания команд, директорий и переменных среды. Вилфредо Санчес [Wilfredo Sanchez], бывший ведущий инженер Mac OS X, работал над *Tcsh* в начале 1990-х в MIT, и неудивительно, что ранние версии Mac OS X поставлялись с *Tcsh* в качестве оболочки по умолчанию; но потом в Panther-версии ОС его заменили на *Bash*.

Употребление псевдонимов-алиасов здесь более гибкое, чем в других оболочках, где не дозволяются никакие подстановки. *Tcsh* с подстановками справляется, поэтому аргументы, переданные псевдониму, можно послать «по цепочке» реальной команде, создавая своего рода мини-функции. Такое вряд ли понравится пользователям *Bash* и *Zsh*: к этому надо попривыкнуть, чтобы осознать, насколько оно полезно.

Основное различие между *Tcsh* и другими оболочками – синтаксис языка сценариев. Как и подразумевается в названии, он напоминает язык C, и хотя циклы *foreach* и *while* действуют схоже, синтаксис совершенно другой, и из-за него скрипты *Tcsh* несовместимы с другими оболочками. В нем также используются конструкции, которые вы не найдете в традиционных скриптах; среди них – безусловно полезный *switch* и сомнительный *goto*.

> «Употребление алиасов более гибкое – в Tcsh разрешена подстановка.»



› *Tcsh*, Умная Оболочка, находит ошибки в написании команд и имен переменных и исправляет их одним нажатием кнопки.

LINUX FORMAT Вердикт

Tcsh
Версия: 6.14
Web: www.tcsh.org
Лицензия: BSD

» *Интересная вариация стандартного синтаксиса оболочки, но на самом деле ничего, превосходящего Zsh и Bash, не предлагается.*

Рейтинг 7/10

Zsh

Продвинутые пользователи, перед вами победитель Bash!

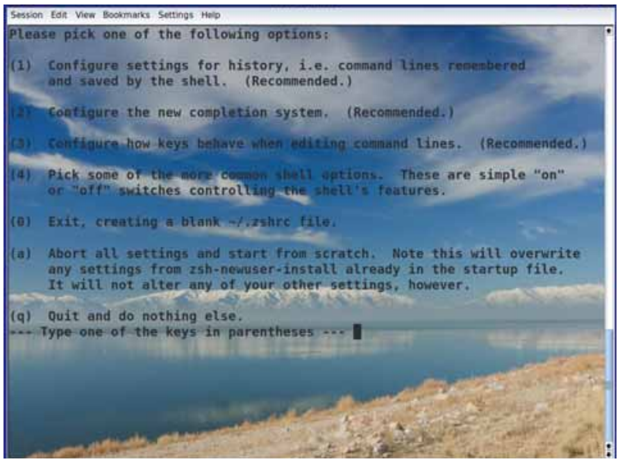
При первом запуске *Zsh* выводится меню настройки, чтобы вы смогли отрегулировать многочисленные мощные и гибкие функции. Все меню содержат массу информации по каждой опции и предлагают разумные значения по умолчанию, поэтому вы можете просто принять их и, если пожелаете, запустить программу настройки позже, когда вам понадобится вся функциональность. *Zsh* снабжен всеми функциями, которые только можно придумать, и даже сверх того – например, проверкой написания.

Zsh и *Bash* предлагают сходный функционал. Обе имеют программируемое дополнение (хотя у *Zsh* оно появилось раньше), встроенные команды и функции для создания скриптов. У *Zsh* также в запасе есть несколько собственных хитростей, например, расширенная подстановка имени файла, которая превращает команду поиска find почти что в ненужное излишество. Включение в путь ** означает соответствие любому символу, включая разделитель-слэш, поэтому **/*.jpg касается всех файлов ***.jpg** в текущей директории и в любых поддиректориях. Мало того, сюда также включаются права доступа к файлу, владелец, тип или отметка времени – большинство опций, предусмотренных find. Например, можно использовать ls -l /**/bin/*(s) для вывода списка всех setuid-файлов в **/bin**, **/usr/bin** и **/usr/local/bin**. Где вы, наши друзья по старым добрым временам Amiga – при наборе имени директории в командной строке *Zsh* переключается на эту директорию. Многие из нас годами приучались ставить **cd** перед именем директории; теперь можно поставить *Zsh* и вернуться к былой лени.

Программируемое дополнение в *Zsh* аналогично той же функции в *Bash*, описанной выше. Можно дополнить аргументы команды, нажав Tab, или включить дополнение имен пользователей или хостов, набрав **ssh**. Поскольку *Zsh* – программируемая оболочка, ее можно расширять сколько душе угодно. Обычный пример – ограничение дополнения имен файлов только теми расширениями, которые способна обработать вводимая команда: так, команда mplayer TAB выведет список всех MPG- и AVI-файлов в текущей директории, подобным же образом обрабатываются и другие формы, с которыми она работает, например, dvd:// и tv://; при этом текстовые документы в той же директории игнорируются.

Без сомнения, *Zsh* – невероятно мощная оболочка. Выполнение скриптов в основном быстрее, чем в *Bash* – по большей части примерно на 20% – однако *Zsh* разработан для интерактивного пользования. Превосходит ли он *Bash*? Почти наверняка, особенно если учесть расширенную подстановку имени файла и более развитую опцию дополнения.

> «Zsh программируемый, можно расширять его сколько душе угодно.»



› **При первом использовании *Zsh* запускается программа настройки. Не пожалейте времени, почитайте про имеющиеся функции и опции.**

LINUX FORMAT Вердикт

Zsh

Версия: 4.3.2
Web: www.zsh.org
Лицензия: ZSH

» *Единственная оболочка в Сравнении, дающая больше функций и гибкости, чем всеобъемлющий Bash. Для знающих пользователей.*

Рейтинг 9/10

Dash

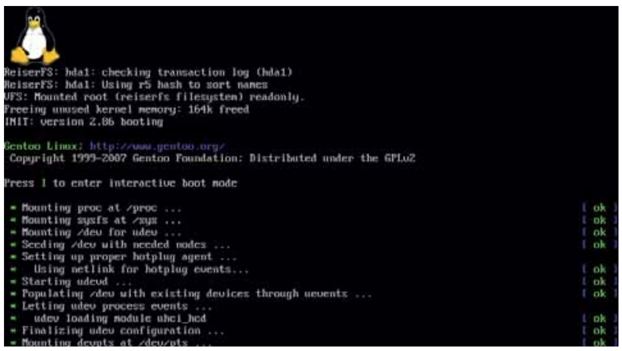
Небольшая оболочка; очень быстро выполняет скрипты.

Если вы ищете небольшую и быструю оболочку, то у *Dash* есть шансы. *Dash* происходит от оболочки FreeBSD под названием *Ash*, и, имея скромный размер 99K, составляет всего одну седьмую от *Bash*. В нем нет тех наворотов, которыми богат *Bash*, но они не всегда и нужны. Как интерактивная оболочка, *Dash* скудноват – здесь нет даже базового tab-дополнения; но он превосходно подходит для выполнения сценариев оболочки POSIX.

Dash почти совместим с POSIX, но кое-какие несовместимости в нем, похоже, имеются, поэтому стоит прочитать документацию и уточнить, какие именно. Однако он сработается с большинством скриптов, использующих **/bin/sh** (любой скрипт, обращающийся к **/bin/sh**, не должен требовать ничего сверх минимума, гарантированного оболочкой POSIX).

Маленький размер также помогает проворности *Dash*. Понятно, что он способствует ускорению и загрузки, и работы скриптов; по сравнению с *Bash* наш тестовый скрипт запустился вдвое быстрее. «Махнув» *Bash* на *Dash* в символьной ссылке **/bin/sh**, вы значительно сократите время загрузки, в течение которой обычно запускается множество сценариев оболочки. Однако, прежде чем это сделать, убедитесь, что у вас не возникнет проблем, способных помешать загрузке. Хороший способ обезопасить себя от внезапных «ударов под дых» – предварительно проверить все init-скрипты вручную.

> «Тестовый скрипт запустился вдвое быстрее, чем в Bash.»



› **Подставив *Dash* в ссылку /bin/sh, вы значительно ускорите загрузку.**

LINUX FORMAT Вердикт

Dash

Версия: 0.5.3.7
Web: http://gondor.apana.org.au/~herbert
Лицензия: BSD

» *Маленькая и быстрая, но не превосходит уровня интерактивной оболочки для нетребовательных систем.*

Рейтинг 6/10

Dsh

Запускайте программы сразу на нескольких ПК.

Dsh – это распределенная оболочка (хотя вообще-то ее название означает *Dancer's Shell*), выполняющая команды сразу на нескольких удаленных машинах. Некоторые рассмотренные здесь оболочки подходят для интерактивного пользования и создания скриптов; а эта – совсем не интерактивна, но зато весьма полезна, когда надо запустить серию команд на группе компьютеров. Альтернативой является нечто вроде

```
for host in node1 node2 node3; do ssh
    $host "my commands"; done
```

С *Dsh*, вы заносите имена хостов в файл **machine.list**, а затем просто набираете Dsh -a my commands. Файл **machine.list** используется опцией -a (или --all), но вы можете поместить любое количество списков имен хостов в групповой директории *Dsh*, и, передав имя списка с помощью опции -g (--group), запустить команду на всех этих машинах.

Опции включают: запуск команд на удаленных хостах последовательно или параллельно, предварение каждой строки вывода именем хоста (что особенно полезно при параллельном запуске команд); и передача команд через стандартный ввод. Команды можно запускать с помощью ssh или rsh, но для должного использования ssh вам понадобится аутентификация по ключу, а для выполнения команды от имени другого пользователя ее нужно будет предварить названием соответствующей учетной записи.

Это – действительно полезный инструмент при работе с группой машин. В нем не хватает только возможности одновременного копирования файлов на несколько компьютеров. Чтобы вытащить файлы с вашего компьютера, вы можете запустить команду scp на других машинах, но это предполагает, что вы предоставите всем им SSH-доступ к вашей рабочей станции – маловероятный сценарий, в особенности если компьютеры-получатели являются общедоступными серверами. Добавка dcp может превратить *Dsh* в отличный инструмент администрирования.

> «Полезный инструмент при работе с группой компьютеров.»



› *Dsh* запускает команду сразу на нескольких ПК: здесь – поверка использования дисковой памяти в основных разделах через сеть.

LINUX FORMAT Вердикт

Dsh

Версия: 0.25.7
Web: www.netfort.gr.jp/~dancer/software
Лицензия: GPL

» *Отличный инструмент удаленного контроля, но сильно нуждается в функции распределенного копирования.*

Рейтинг 7/10

Sash

Желаем, чтобы она вам не понадобилась!

Sash – это просто самая крупная из рассмотренных здесь оболочек, и не без причины. Ее название – сокращение от *Stand Alone Shell* (Оболочка-одиночка), и в ней содержится несколько встроенных команд, которых нет ни в одной другой оболочке. Когда мы скажем, что эти команды – *mount*, *ls*, *cp*, *tar* и *gzip*, вы поймете, для чего нужен *Sash*. Это – оболочка для восстановления системы, которая работает даже тогда, когда большая часть вашей файловой системы утрачена. Если **/boot** у вас находится на отдельном разделе, и вы не монтируете его при загрузке, или монтируете в режиме только для чтения, поместите на него копию *Sash*, и выдержите даже rm -fr /*. Используя встроенные команды *Sash*, вы сможете восстановить свои резервные копии (у вас ведь есть резервные копии, правда?) и запустить работу снова, не теряя времени. После отказа системы восстановление возможно даже при отсутствии важных библиотек: все необходимое – это ядро и *Sash*.

По причине ограничений на размер, команды, встроенные в *Sash*, являются урезанными версиями своих аналогов, поэтому *Sash* предоставляет доступ и к тем, и к другим. Для запуска стандартной команды *ls* вводите ls, а -ls вызовет встроенную. Стандартная команда недоступна? Сделайте псевдоним ls, разворачивающийся в -ls. Если вас угораздило полностью удалить **/bin**, используйте *aliasall*, и все команды будут выбираться из встроенных. Того же можно добиться, запустив *Sash* с опцией -a. Команда *help* выведет список встроенных команд и информацию по использованию любой из них.

Sash также умеет запускать скрипты с помощью опции -f, а значит, вы можете автоматизировать даже процесс восстановления. Если безобразия у вас творятся часто – настолько, чтобы вам захотелось автоматизировать использование *Sash* для этих целей – возможно, стоит пересмотреть и изменить свой подход к работе. *Sash* вряд ли подходит на роль стандартной интерактивной оболочки, но он из тех приложений, наличие которых под рукой приведет вас в состояние дикой радости, если с вами стряслось то, что бывает только у других.

> «Хотите автоматизировать Sash? Лучше пересмотрите подход к работе!»



› *Sash* спасет ваше добро при порче корневой файловой системы, но не ждите, пока грянет гром: установите его прямо сейчас.

LINUX FORMAT Вердикт

Sash

Версия: 3.7
Web: www.canb.auug.org.au/~dbell
Лицензия: Свободно распространяемая

» *Ограниченный набор функций, для повседневного использования не годится, но незаменима в случае аварии.*

Рейтинг 5/10

Оболочки

Вердикт

Zsh 9/10

Группа программ, представленных в данном **Сравнении**, довольно необычна, потому что в ней есть программа, уже установленная практически в каждом дистрибутиве.

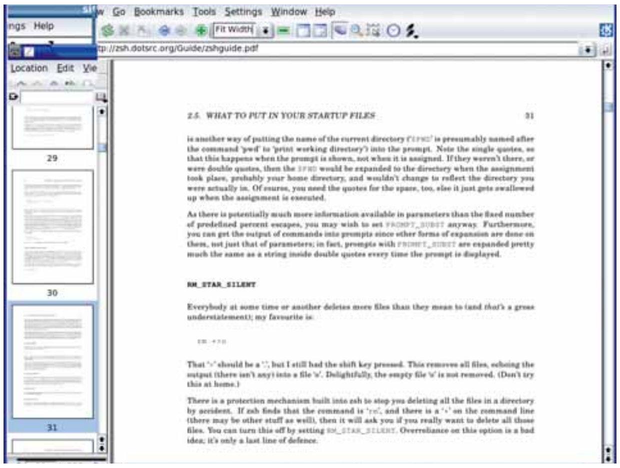
Вышло так, что *Zsh* победил фактически по всем параметрам: он быстр, достаточно компактен, имеет превосходную документацию и лучшее дополнение, чем любая из представленных здесь оболочек. Если вы прежде не использовали tab-дополнение в оболочке, попробуйте: время здорово экономится. При интерактивном использовании *Zsh* непревзойденна, благодаря своим функциям и дружелюбию ко всем, кто познакомился с документацией. Нужны еще аргументы? Что ж, автор уже сделал *Zsh* оболочкой по умолчанию вместо *Bash*.

Мы не говорим, что *Bash* – плохая оболочка, отнюдь, просто *Zsh* все делает чуть лучше. *Bash* по-прежнему остается эталоном, по которому оценивают остальных, возможно, именно поэтому он столь распространен, и действительно устанавливает высокую планку. Если вы пишете скрипты оболочки посложнее, чем простой список команд, стоит воспользоваться именно *Bash*, и именно поэтому дистрибутивы выбирают его оболочкой по умолчанию.

Tcsh в известной мере является анахронизмом. Конечно, его ни в коем случае нельзя назвать плохой оболочкой, но нестандартный синтаксис несколько уменьшает ценность *Tcsh* для написания сценариев, учитывая еще и тот факт, что это ничем не компенсируется. Синтаксис *Tcsh*, возможно, упрощает создание скриптов, но поскольку запускать их можно будет только на своей машине, вряд ли это по-настоящему полезно.

Dash примечателен небольшим размером и быстротой, но более, собственно, ничем. Он пригодится тем, кому надо поторопить медленное оборудование. *Dsh* – не похож на остальные, но может стать мечтой сетевого администратора – если кроме выполнения команд даст возможность распределенного копирования файлов.

Fish буквально раскрывает глаза на то, какой дружелюбной может быть оболочка. Его настройки по умолчанию (исключая странный выбор программы просмотра помощи по умолчанию) просто великолепны для начинающих пользователей, желающих привыкнуть к работе с командной строкой. Он нарушает совместимость со стандартными сценариями оболочки, но благодаря этому приобретает гибкость, обеспечивающую более человечный синтаксис, и к тому же он никогда и не предназначался для запуска скриптов.



› Видите – мы говорили что в Zsh есть все, что может вам понадобиться: даже в его PDF-руководстве целых 355 страниц!

Sash заслуживает особого упоминания, несмотря на низкий рейтинг. Он заработал 5 из 10 только потому, что в сравнении с остальными участниками предлагает слишком мало, за исключением одного: ему вообще ничего не надо, чтобы запуститься. Никому и в голову не придет использовать его в качестве стандартной оболочки, но *Sash* надо держать в надежном месте; возможно, когда-нибудь он станет вашей палочкой-выручалочкой. LXF

Как по-вашему?

Вы пробовали работать с разными оболочками, или вполне довольны стандартной? Вас пугает работа с командной строкой? Если да, то не поможет ли вам решиться более дружелюбная оболочка? Присылайте свои отзывы на наше **Сравнение**: letters@linuxformat.ru

> «Zsh быстр, компактен, имеет превосходную документацию и самое лучшее дополнение.»

Краткая история оболочек

В начале был *sh*, известный также, как Оболочка Томпсона (*Thompson Shell*) – оболочка, появившаяся вместе с первым Unix в 1971 г. Он предназначался только для выполнения команд, не для скриптов. Шесть лет спустя его вытеснила Оболочка Борна (*Bourne Shell*), которую тоже назвали *sh*. Обе оболочки вышли из *Bell Laboratories*. В *Bourne Shell* появились скрипты и прочие инновации, например, командная подстановка, использующая обратный апостроф (VAR=`command --arg`).

Оболочка Корна (*Korn Shell*, *ksh*) также была разработана в Bell Labs (может, AT&T – их владелец – на самом деле корпорация оболочек?). Она была совместима с *Bourne Shell*, и в ней добавилось еще больше функций, включая расширенные скрипты и такие вещи, как предыстория (журнал) команд. Если вы когда-нибудь работали с оболочкой без журнала команд – как изначальная оболочка Amiga, например – вы поймете, какой это был гигантский скачок.

Bash основана на *Bourne Shell*, а на создание *Zsh* вдохновила *Korn Shell*, но благодаря стандарту POSIX оба отлично сотрудничают.

Таблица сравнения функций

	Лицензия	Размер исполняемого файла [1]	Tab-дополнение	Расширенное дополнение	Детализаация	Онлайн-справка	POSIX-совместимость	Проверка написания
Bash	GPL 2	41k	✓	✓	✓	✓	✓	☒
Dash	BSD	80k	☒	☒	☒	✓	☒	☒
Dsh	GPL 2	21k	N/A	N/A	✓	☒	N/A	☒
Fish	GPL 2	318k	✓	✓	✓	✓	☒	☒
Sash	Свободно распространяемая	971k	✓	☒	✓	✓	☒	☒
Tcsh	BSD	326k	✓	☒	✓	☒	☒	✓
Zsh	ZSH	511k	✓	✓	✓	☒	✓	✓

[1] Откомпилированного GCC 4.1.2 на Athlon64.

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.



ЛАДИСЛАВ БОДНАР
основатель, начальник, редактор и сотрудник DistroWatch.com.

Linux на палочке

Поразительно, как простая идея порой оборачивается грандиозным успехом. Возьмем Mandriva Flash, полноценную ОС на загрузочном USB-носителе. С тех пор как французская фирма Linux начала продавать элегантный USB-брелок, содержащий Mandriva 2007, ему было посвящено множество обзоров, что даже вылилось в сообщения «временно отсутствует на складе», не так давно висевшие в онлайн-магазине.

Что же особенного в дистрибутиве Linux, втиснутом в 4-ГБ флэш-драйв? Во-первых, в отличие от Live-CD, на USB-диске пользователи могут сохранять настройки и данные. Во-вторых, производительность системы на удивление хороша, особенно при подключении к порту USB 2.0. И в-третьих, тот, кто его купит, заодно получит устройство, полезное даже тогда, когда ПО на нем устарело.

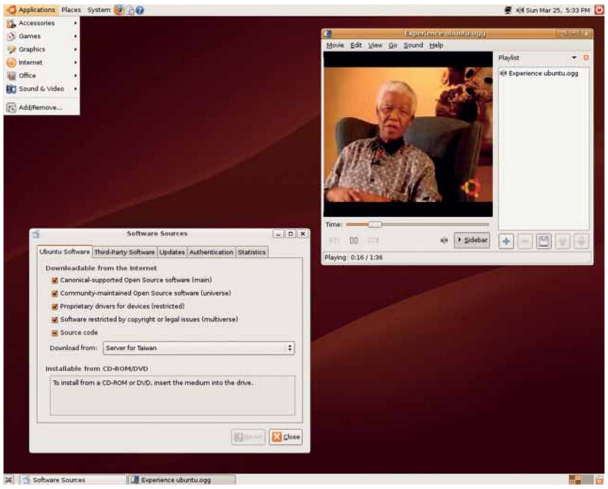
Прибыльное дело

Последнее, вероятно, и сделало Mandriva Flash бестселлером. Признаемся: многие из нас неохотно платят за программы, доступные бесплатно в Интернет,е но мы не раздумывая раскошелимся на устройство, особенно если оно классное, содержит Linux и подвешивается на колечко от ключей вместе с нашими данными. Ждите всплеска популярности идеи Mandriva Flash в конце этого года. Многие сообщества дистрибутивов пытаются найти способ изыскать средства для покрытия стоимости своей работы. От продажи свободных программ и просьб о пожертвованиях прибыли немного, но загрузочный USB-брелок с предустановленным Sabayon Linux, Linux Mint или Zenwalk Linux – предложение достаточно привлекательное, чтобы рука сама потянулась щелкнуть на кнопке Купить.

ladislav.bodnar@futurenet.co.uk

Юный олень

Ubuntu Feisty Fawn 7.04 Настольный дистрибутив с улучшенным установщиком и новым инструментом миграции.



› В Ubuntu 7.04 (да, опять коричневый!) легко добыть программы и добавить поддержку мультимедиа.

Большинство читателей, следящих за онлайн-зоной Linux, согласятся: Ubuntu за последние месяцы привлекал к себе больше внимания, чем ему бы полагалось, как резюмировал Кори Доктороу [Cory Doctorow] на BoingBoing.net, отказавшись от OS X ради Ubuntu в октябре 2006 г.

Это внимание нельзя назвать незаслуженным. Другие авторы дистрибутивов до последнего времени придерживались мнения, что Linux не готов для среднего пользователя, тогда как разработчики Ubuntu внедряли свежие идеи, чтобы превратить его в серьезного соперника на рынке настольных операционных систем. Feisty Fawn [Храбрый Олененок] – новейший стабильный релиз дистрибутива – предоставил больше веских доводов для перехода на Linux, чем любой другой дистрибутив.

Возьмем новый инструмент миграции. В процессе начальной установки системы установщик проверит ваш жесткий диск на наличие других ОС и предложит перенести их настройки, данные и файлы в Ubuntu. Для тех пользователей, на чьих жестких дисках установлена Windows, улучшенный установщик Ubuntu облегчит перенос в новую ОС закладок браузера, электронных писем и контактов интернет-пейджера. После перезагрузки пользователи обнаружат, что их самые важные данные уже доступны, без лишней суеты.

> «Намерен стать самым инновационным Linux-дистрибутивом этого года.»

Другим важным улучшением Ubuntu 7.04 является новая опция установки (по требованию) популярных – но не свободных и, возможно, патентованных – медиа-кодеков. Например, если пользователь попытается воспроизвести MP3-файл, поддержка которого не включена в стандартную установку Ubuntu, то всплывающий диалог предупредит пользователя об этом и предложит загрузить необходимые компоненты, чтобы сделать воспроизведение возможным.

Разработчики Ubuntu нашли прекрасное решение спорного вопроса о предоставлении проприетарных драйверов для видеокарт Nvidia и ATI и другого оборудования. Вместо их установки по умолчанию, Ubuntu сигнализирует об их существовании посредством иконки в системном лотке сразу после первого входа пользователя. Таким образом, пользователь сам может принять осознанное решение: использовать ли драйвер с расширенной функциональностью, типа поддержки 3D-эффектов рабочего стола, или оставить полностью поддерживаемый открытый драйвер, но с ограниченными возможностями.

Среди других новшеств: Plug-and-Play использование сетевых ресурсов, автоматически настраивающее доступ к другим машинам в локальной сети, активация 3D-эффектов рабочего стола одним щелчком мыши и значительно уменьшенное время загрузки и выключения. Хотя в момент написания статьи он был все еще в стадии тестирования, Ubuntu 7.04 определенно намерен стать одним из наиболее важных и инновационных релизов дистрибутивов Linux этого года. Даже если вам не по душе всеобщий интерес к Ubuntu и продолжающаяся шумиха в средствах массовой информации, несомненно одно: с каждым новым релизом проект делает шаг к устранению различий в использовании проприетарных операционных систем и Linux.

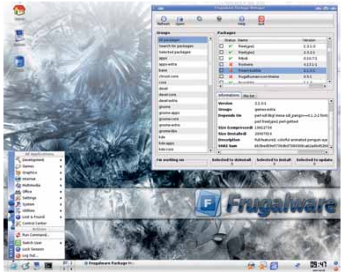
www.ubuntu.com

Милый скромник

Frugalware Linux 0.6 Любопытный расширяемый дистрибутив из Венгрии.

Иногда, исследуя дистрибутивы вне первой двадцатки DistroWatch.com, натыкаешься на удивительный сюрприз. Такой, как Frugalware Linux [frugal – бережливый, англ.], очень экономный дистрибутив – его создал Миклош Вайна [Miklós Vajna] с группой молодых разработчиков из Будапешта, Венгрия. Сейчас, будучи трех лет от роду и пройдя 6 стабильных релизов, Frugalware стал скромной, но приятной настольной системой, ориентированной на среднего пользователя.

Отчего же Frugalware достоин внимания? Очень интересный аспект – его управление программными пакетами. Frugalware не только стремится предоставить последнее и лучшее на момент выпуска, но также использует быстрый и гибкий инструмент управления пакетами – Pacman, изначально разработанный для Arch Linux. Кроме упрощения установки любого из около 3 000 пакетов, доступных в онлайн-репозиториях дистрибутива, он также позволяет создавать двоичные пакеты для Frugalware прямо из исходного кода.



› **Пакетами во Frugalware управляет графическая утилита на базе PyGTK.**

Новички Linux могут отклонить этот дистрибутив из-за отсутствия графического установщика и других милых утилит настройки, но более продвинутые сочтут Frugalware ясной, простой, быстрой и мощной ОС, оптимизированной для процессоров i686 (отдельное издание для x86_64 также имеется). Web-сайт проекта имеет все, что должен любой хороший дистрибутив: списки рассылки и форумы, каналы IRC-обсуждений, документацию в стиле wiki, советы по безопасности, блоги разработчиков и даже регулярные рассылки.

www.frugalware.org

Будь настороже

BackTrack 2.0 Live CD для параноиков: дистрибутив, пробующий все на прочность.

Компьютерная безопасность – скучная тема. Готовить отпор каждому злодеянию, намеренному нарушить мирное сосуществование миллионов компьютеров в сети Интернет – занятие и нудное, и трудное. Но принятие мер по безопасности не обязано быть тяжким трудом. Как недавно показала швейцарская фирма Remote-Exploit, с BackTrack 2.0 это даже забавно.

BackTrack – специализированный LiveCD, основанный на Slackware и содержащий огромную коллекцию инструментов безопасности для управления сетью, тестов на проникновение, определения уязвимостей, цифрового права, усиления привилегий, реинжиниринга и решения других задач. Инструменты, опрятно расположенные в меню запуска KDE, определят потенциальные проблемы безопасности компьютерной системы, а также помогут с анализом ситуации, если система взломана.

Предоставление столь большого числа мощных инструментов в одном компактном пакете, однако, и опасно. Попав в дурные руки, CD может быстро превратиться в мощный инструмент взлома, способный легко вскрыть системы с известными уязвимостями или неоптимальными настройками безопасности. Но это плата за то, что создатели называют «агрессивной безопасностью» – знание ПО, которое злобные кракеры используют в своей гнусной деятельности, может помочь всем нам.



› **BackTrack 2.0 содержит около 300 утилит безопасности.**

Remote-Exploit также предлагает тренинги, публикует статьи по компьютерной безопасности и многое другое. Но BackTrack LiveCD – самый ценный их продукт. Он попросту один из лучших инструментов безопасности на рынке. LXF

www.remote-exploit.org

Дистрибутивы на базе Ubuntu

Число создателей дистрибутивов, которые сочли Ubuntu идеальной основой для своего детища, неуклонно растет. В большой степени это связано с предсказуемым циклом разработки Ubuntu – каждые шесть месяцев проект синхронизирует свои программные пакеты с нестабильным репозиторием Debian, затем использует свой процесс разработки для обновления наиболее выдающихся программ до последних версий, исправляет все ошибки и стабилизирует код. Стабильный релиз Ubuntu затем может служить отправной точкой для других, более специализированных решений, как показано в таблице ниже.

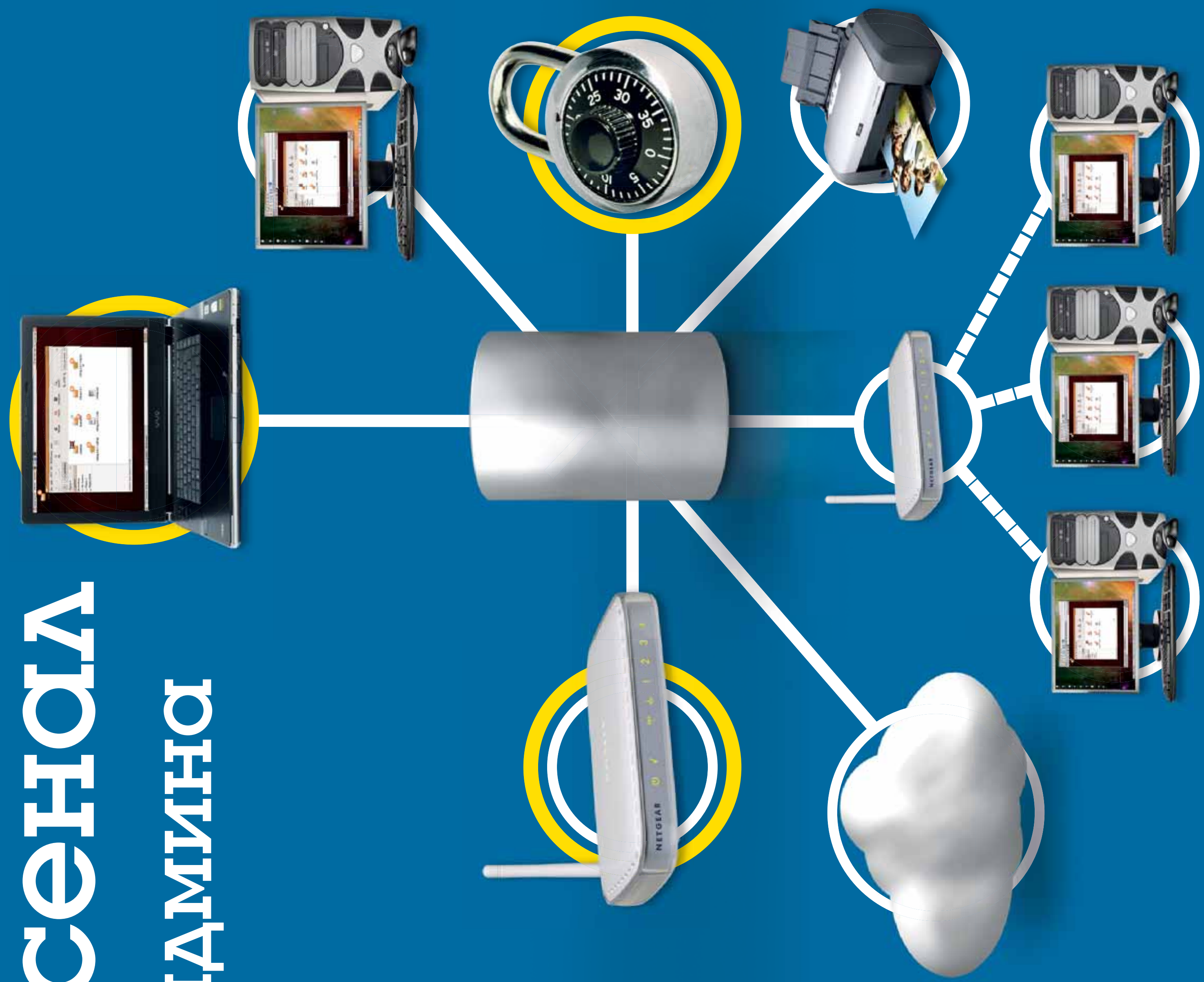
Название	Цель
SimplyMepis	Дружественный к новичкам, преднастроенные модули устройств и модули расширения.
Linux Mint	Поддержка популярных медиа-форматов «из коробки».
Freespire	Дружественный к новичкам, поддержка популярных медиа-форматов, на ранней стадии.
Ubuntu Christian Edition	Слегка измененный Ubuntu с интегрированным ПО для изучения Библии и тщательно настроенным брандмауэром.
Nexenta	Ядро OpenSolaris вместо Linux, программные пакеты Ubuntu.
Ulteo	Основанный на KDE дистрибутив от Гаэля Дюваля (экс-Mandrake), на ранней стадии.
Pioneer Linux	Для сборщиков систем, чтобы устанавливать несвободные приложения, с KDE и Automatix.
GNewSense	Ubuntu минус «двоичные куски» – рекомендован FSF.
Fluxbuntu	Легковесный Ubuntu с Fluxbox – хорош для старых компьютеров.

Хит-парад дистрибутивов

10 самых посещаемых страниц на DistroWatch.com, с 28 февраля по 4 апреля 2007 г. (среднее число визитов в день)

Дистрибутив	Число визитов	
1 Ubuntu	3,327	‹›
2 PCLinuxOS	2,145	↑
3 SUSE	1,785	↓
4 Fedora	1,489	↑
5 Sabayon Linux	1,473	↑
6 SimplyMepis	1,258	↑
7 Mandriva	1,037	↑
8 Debian/GNU Linux	1,027	↑
9 Mint	939	↑
10 Damn Small Linux	817	↓

» DistroWatch.com оценивает популярность дистрибутивов, основываясь на числе визитов на страницу каждого дистрибутива. Хотя это не дает представление о действительном числе установок, но показывает, какие дистрибутивы более популярны за определенный промежуток времени.



Арсенал сисадмина

Вы можете быть великодушным надзирателем за одной машиной или безумным оператором на тысяче, но у вас всегда есть шанс научиться чему-нибудь новому. **Грэм Моррисон** накопил отличные советы по улучшению производительности, безопасности системы, а также работе с сетью.

LINUX FORMAT На DVD

- Dban 1.0.7
- GParted 0.3.3
- InitNG 0.6.10
- DenyHosts 2.6
- BackupPC 3.0.0
- SystemImager 3.7.6
- Lire 2.0.2
- Fwlogwatch 1.1
- Squidalyser 0.2.55
- Analog 6.0
- logcheck 1.1.1
- PortSentry 1.2

Американский народный певец Эни Ди Франко [Ani DiFranco] сказал однажды, что каждый человек – актер. Ну, а каждый пользователь Linux – системный администратор. Если вы не просто шастаете в Интернете да читаете электронную почту, то вам, несомненно, приходилось использовать командную строку, устанавливать дополнительные пакеты и связывать свой компьютер с остальным миром через сеть.

Это – задачи системного администратора, и существует масса способов упростить его труд. Вот мы и свели вместе лучшие подсказки и советы, какие только смогли накопать, чтобы помочь вам в администрировании вашей Linux-системы, будь то одна машина или целая сеть компьютеров. А если системное администрирование – ваша работа, вы найдете здесь кучу такого, что поможет вам получить повышение.

Мы разделили статью на три части. Первая часть отведена малым сетям, состоящим менее чем из пяти машин. Большинство советов пригодится и для одной машины, и уж несомненно они будут полезны в случае небольшой домашней сети. Вторая часть немного более амбициозна, она касается уже сетей масштаба малого бизнеса. А на закуску мы припасли несколько рекомендаций для грандиозных Linux-сетей, которым не обойтись без небольшого IT-отдела. Чтобы вам было проще найти требуемый материал, мы также разбили все советы на три категории: работа с сетью, безопасность и производительность.

Если вы планируете добавить в вашу домашнюю сеть пару компьютеров, покончить со старой проприетарной системой в офисе или просто запустить виртуальную машину, здесь вы узнаете, как сделать это быстрее, безопаснее, и, главное, проще. Наши подсказки могут войти в рабочий арсенал системного администратора, и снабжены подходящими утилитами, которые мы разместили на диске этого месяца (слева).

Сети менее чем из пяти машин



Безопасность: Управление пользователями



Если вы у своего компьютера – единственный и неповторимый, вам нечего опасаться, что кто-то влезет в ваши владения. Но стоит добавить новых пользователей – и прощай, покой. Большинство дистрибутивов включает панель управления пользователями, позволяющую добавлять или удалять их из системы. Она также позаботится о создании домашнего каталога, а зачастую и файлов шаблонов рабочего стола. Однако на вашей ли машине располагаются эти пользователи, или же на других машинах под вашим надзором – все, что им можно и чего нельзя, определяется тем, членами какой группы они являются.

Каждый дистрибутив Linux использует группы немного по-своему, но теория у них одинаковая. Взглянув на файл **/etc/group**, можно увидеть список из десятков групп, и некоторые сопровождаются несколькими именами пользователей. Вы можете найти группу 'cdrom', рядом с которой будет список всех пользователей, являющихся членами этой группы и, следовательно, имеющих доступ к CD-приводу в вашей системе. Чтобы отнять право доступа, просто удалите имя пользователя из этой группы.

То же самое можно сделать более изящно при помощи графического менеджера пользователей вашего дистрибутива. Права доступа к файлам и каталогам включают установки для группы, так что даже если пользователь не является владельцем файла, он может иметь доступ к нему, если он – член соответствующей группы. Используйте файловый менеджер для изменения групповых прав и ограничения доступа.

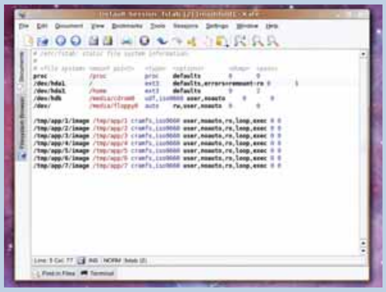


› **Члены группы 'admin' в Ubuntu могут пользоваться *sudo*.**

Производительность: Переход на журналируемую файловую систему



Если на вашей машине все еще присутствует старая файловая система ext2, вы можете получить значительное повышение производительности, если перейдете на ext3. Чтобы проверить, какие файловые системы используются в разных разделах вашей системы, откройте **/etc/fstab** – файловая система указана там в третьем столбце (под 'type'). Использование ext3 создает сразу несколько преимуществ. Дело не только в лучшей производительности, но еще и в том, что ext3 – журналируемая файловая система: каждое обращение к жесткому диску заносится в «журнал». В случае сбоя журнал можно использовать для восстановления файловой системы в прежнее работоспособное состояние – будем надеяться, без потери данных.



Первый шаг при преобразовании ext2-раздела в ext3 состоит в создании журнала как такового. Для этого, действуя от имени системного администратора, введите в командной строке **tune2fs -j /dev/hda1**. Вместо **hda1** подставьте имя устройства для вашего раздела, указанное в файле **/etc/fstab**; операцию можно проделать как для смонтированного, так и для размонтированного разделов. Последний шаг – заменить в файле **/etc/fsab** тип файловой системы на 'ext3'.

› **Файл /etc/fstab описывает, какие файловые системы подключены.**

Работа с сетью: Туннелирование портов через SSH



Ключевой момент безопасности сервера – держать как можно меньше открытых интернет-портов. Но блокирование доступа к вашим серверам из Интернета посредством брандмауэра еще не значит, что будут они недоступны для вас. Используя SSH-соединение, вы можете туннелировать порты, которые использует ваш сервер в удаленной локальной сети, на порт машины, за которой сидите. Это – великолепное решение, если нужно, например, получить доступ к web-серверу, находящемуся за брандмауэром.

Буква 'L' – вот разгадка этого волшебства. Использование опции **-L** в командной строке SSH свяжет удаленный порт с локальным. Например, если вам нужно пробросить порт протокола удаленного доступа к рабочему столу (обычно это 5901), введите следующее:

```
ssh -L 5901:localhost:5901 remote_ip
```

После этого вы сможете обращаться к удаленному компьютеру, подключая VNC-клиент к порту 5901 локального компьютера. При этом данные будут переданы через SSH-соединение. Вам следует только иметь на брандмауэре открытый SSH-порт (обычно, 22); кроме того, при такой передаче данные не могут быть подделаны.

Использование Live CD для сохранения системы

Будучи пользователями Linux, мы находимся в благоприятном положении, ибо имеем выбор из нескольких различных Live CD, позволяющих вылечить захворавшую систему. Live CD – идеальное решение для этой задачи, так как зачастую достаточно одного диска со специальными утилитами и приложениями, необходимыми для реанимации вашего компьютера. Даже некоторые восстановительные утилиты Windows применяют Linux Live CD для загрузки поврежденной системы «от Microsoft».

Вот три лучших:

» **Darik's Boot and Nuke** Маленький 2-МБ ISO-образ, который можно загрузить с CD или даже с дискеты. Dban тщательно удалит все данные на жестких дисках компьютера, и всем, кто позже получит к нему доступ, будет фактически невозможно узнать, что содержалось на этой машине. Идеальный вариант, если вы хотите подарить ваш компьютер, отдать его в другой отдел или попросту выбросить. http://dban.sourceforge.net

» **GParted** Один из наиболее популярных Live CD, поскольку выполняет те же задачи, что и пакеты, стоящие ощутимых денег. Позволяет создавать новые разделы на вашем жестком диске, менять размеры уже существующих и форматир
менных файловых систем. Это также подходящий инструмент для восстановления, если вам не удается запустить вашу ОС, поскольку загрузочное меню настроено так, чтобы позволить вам загрузиться с любого доступного диска. http://gparted.sourceforge.net

» **SystemRescueCd** Как дистрибутив Linux «все-в-одном», со всевозможными утилитами для оживления вашей системы, SystemRescueCd сложно превзойти. Наряду с редакторами разделов и программами создания дисковых образов, настройщиками сети и тестерами оборудования, имеется также целый выводок утилит для починки файловых систем. Все это, плюс удобный рабочий стол Linux, означает, что диск предлагает наилучшую возможность излечить компьютер при помощи технологии Live CD. www.sysresccd.org



Сети менее чем из пяти машин (продолжение)

Производительность: Ускорение загрузки системы



Ожидание появления рабочего стола Linux бывает тягостным, особенно если вам нужно всего лишь проверить почту. Именно по этой причине многие из нас упорно отказываются выключать компьютер в обеденный перерыв или вечером. Для решения этой проблемы в Ubuntu Edgy Eft была введена система Upstart, заменившая init, но мы почти не заметили прироста в скорости запуска.

Проблема загрузки Linux состоит в том, что первый процесс, запускающий все остальные, делает это последовательно, и когда очередной процесс ждет какого-либо события, остальные процессы стоят. Вроде улицы с односторонним движением: приходится ждать, пока сетевым процессам перед вами присвоят IP-адрес. Решение – выполнять как можно больше процессов параллельно. К этому-то и стремится InitNG, гордо назвавшийся «процессом инициализации нового поколения»: при начальной загрузке он способен запускать параллельно несколько процессов. InitNG выполняется на очень низком уровне, так что рекомендуем использовать готовый пакет для вашего дистрибутива и устанавливать его только через пакетный менеджер. Не считая Ubuntu, пакеты InitNG имеются для всех распространенных дистрибутивов.

Производительность: Быстрый запуск приложений



Если скорость загрузки вашей системы – это повод взгрустнуть, то время запуска некоторых приложений определенно усугубляет проблему. И мы имеем в виду не только *OpenOffice.org*. Если сложить все те секунды, которые мы что ни день тратим на ожидание – хоть там браузера, хоть почтового клиента – времени вполне хватит заварить чашку чая и вытащить из коробки печенье.

Вы удивитесь, но можно кое-что сделать для ускорения этого процесса (мы не про чай: хорошая заварка спешки не допускает). Утилита *prelink* ускоряет запуск приложений. Большую часть времени процедура загрузки приложения занимается поиском разделяемых библиотек, образующих массивную сеть зависимостей, и зачастую одни библиотеки базируются на других. Библиотеки компонуются с исполняемым файлом в процессе старта вашего приложения. *Prelink* определяет зависимости заголовка, тем самым урезая повторный процесс их вычисления. Установите пакет *Prelink* для вашего дистрибутива и откройте в текстовом редакторе **/etc/default/prelink** – вам следует только убедиться, что там содержится строка 'PrelinkING=yes'.

Prelink также установит небольшой скрипт, запускающийся ежедневно из **/etc/cron.daily/prelink**. Вам потребуется вручную запустить его один раз, после установки, для инициализации базы данных *Prelink*; поработает он долгонько, но потом ежедневных проверок вы даже и замечать не будете. *Prelink* уменьшил время запуска нашего *OOo Writer* с 25.027 секунд до 8.943.

Безопасность: Требуйте приличных паролей



Ненадежные пароли пользователей чреваты серьезной опасностью для вашей системы. Если пользователей не слишком много, то при локальной работе проблем не будет. Однако большинство из нас использует SSH для удаленного соединения, а при этом учетные записи ваших пользователей постоянно находятся под угрозой со стороны злонамеренных машин. Тут мало простой просьбы придумывать пароли получше: необходимо принудить их быть надежными. К счастью, простой заменой парольного модуля (Pluggable Authentication Module, PAM), используемого по умолчанию, на более строгий можно заставить ваших пользователей выбирать пароли правильно. PAM используется в качестве механизма аутентификации для всех видов ежедневных задач, включая проверку пароля при входе в систему. Модуль, который вам нужен, называется Passwdqc, и его следует установить через менеджер пакетов вашего дистрибутива. После этого останется только изменить настройку PAM, чтобы использовать новый модуль вместо старого. Откройте файл **common-password** (он обычно находится в каталоге **/etc/pam.d**), раскомментируйте строку, начинающуюся с 'password', и вставьте вот что:

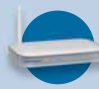
```
password required pam_passwdqc.so
  enforce=users
```

Готово. Новый модуль даже будет сам предлагать надежные пароли вашим пользователям.



› PAM заставит пользователей выбирать безопасные пароли.

Работа с сетью: Разделите файлы с пользователями Windows.



Если в вашей сети несколько машин, есть вероятность того, что некоторые из них работают под Windows. А тогда вам, наверное, захочется использовать файлы совместно.

Это – задача для *Samba*. Она включена во все основные дистрибутивы, но серверная часть, необходимая, чтобы пользователи Windows видели ваши файлы, устанавливается редко. Используя менеджер пакетов вашего дистрибутива, инсталлируйте пакеты с именами **samba** или **samba-server**. Многие дистрибутивы заодно установят панель управления для создания *Samba*-пользователей и настроят *Samba* для совместного доступа только к домашнему каталогу пользователя, так что ему понадобится учетная запись и на Windows- машине, и на сервере Linux. Удобно иметь одинаковые имя пользователя и пароль, хотя можно предусмотреть, чтобы Windows требовала их ввода при соединении. Чтобы Linux-машина появилась в сетевом окружении Windows, надо убедиться, что имя рабочей группы, определенное в файле конфигурации *Samba* (**/etc/samba/smb.conf**), совпадет с используемым в Windows.



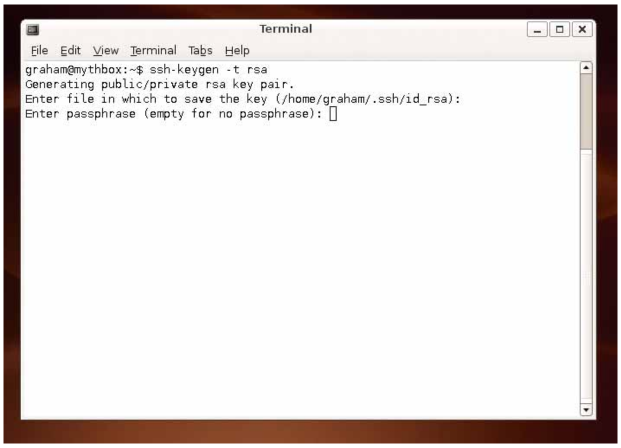
› Многие дистрибутивы имеют собственную панель управления *Samba*, чтобы делиться файлами с пользователями Windows.

Безопасность: SSH без паролей



Мы уже писали, что использование SSH без паролей очень удобно. Здесь, в *Linux Format*, SSH применяется практически везде, будь то простой удаленный доступ, управление локальными учетными записями, безопасная передача файлов по SFTP или туннелирование портов с удаленного узла на локальный. SSH жизненно важен для управления Linux-сетью, и удаление пароля из ежедневной рутинной работы не только упростит управление ею, но и устранит самую большую угрозу для безопасности вашего SSH-сервера: подбор пароля. «Беспарольный» SSH использует пару ключей: один – общедоступный, а другой – закрытый. Вопреки интуиции, именно клиент хранит закрытый ключ, а сервер использует открытый ключ для подтверждения личности клиента (единственного обладателя закрытого ключа).

Вы можете создать как закрытый, так и открытый ключи, введя **ssh-keygen -t dsa**. Вас попросят ввести парольную фразу. Если вы ничего не введете, соединение будет беспарольным, но если вам небезразлична безопасность, то использование парольной фразы в комбинации с парой открытый-закрытый ключ – это самая высокая степень защиты, которую только может дать SSH. В завершение следует скопировать файл **id_dsa.pub** из каталога **~/.ssh** в **~/.ssh/authorized_keys** на SSH-сервере. Оба каталога **.ssh** обязаны иметь права доступа 644 (**chmod 644 ~/.ssh**), иначе SSH откажется работать, выдав маловразумительное сообщение.



› SSH без пароля – риск для безопасности, но в надежно защищенной сети это позволит упростить многие задачи.

Советы

Вильям фон Хаген [William von Hagen] выбрал свой любимый совет из своей книги Linux Server Hacks, Том 2.

«Совет 94, «Восстановление данных с поврежденных дисков», возможно, мой самый любимый, потому что нет ничего трагичнее потери данных», – говорит фон Хаген. «Резервные копии обычно слегка отстают от времени (но все их делают, правда?). Здесь обсуждаются утилиты восстановления/исправления для разных типов файловых систем и дается несколько дельных подсказок по починке файловых систем ext2/3. Затронут также вопрос клонирования наиболее поврежденных дисков с использованием улучшенной версии dd [*ddrescue*, http://ftp.gnu.org/gnu/ddrescue].»

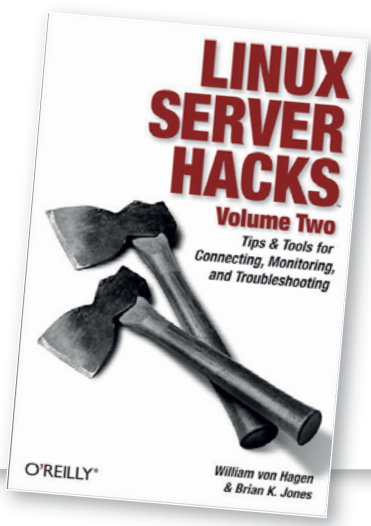
«Прежде всего совет касается дисков, содержащих файловые системы, а не физические или логические тома, но с ними бы справился тоже. Способность восстанавливать данные через этот механизм отражает такие основы Unix/Linux, как доступ непосредственно к устройству и принцип «все является файлом» – и нет ничего милее, чем вновь созерцать свои данные, обретенные после сбоя накопителя.»

Если в файловой системе возникла проблема, вы обычно начинаете видеть загадочные сообщения об ошибках при каждом использовании определенного диска или его раздела (взгляните в **/var/log/messages**, если думаете, что вы что-то упустили). Фон Хаген советует первым делом проверить целостность файловой системы, используя непопулярную команду *fsck*, однако вначале вам следует получить доступ к отмонтированной файловой системе. Самый простой способ достичь этого – загрузиться с Live CD. Ваша версия команды *fsck* зависит от формата вашей файловой системы, но большинство современных дистрибутивов используют *fsck.ext3* для файловой системы ext3, более старые версии используют либо *fsck.ext2* для *ext2*, либо *fsck.reiserfs* для ReiserFS. Например, команда **fsck.ext3 /dev/hda1** проверит первичный раздел первого IDE-диска, отформатированного под *ext3*.

Затем фон Хаген объясняет, как воссоздать суперблок диска. «Суперблок содержит основные сведения о файловой системе, включая первичные указатели на блоки с информацией о файловой системе (называемые индексными дескрипторами)», говорит он. «К счастью, когда вы создаете файловую систему ext2 или ext3, соответствующая утилита (*mke2fs* или символическая ссылка на нее под именем *mkfs.ext2* или *mkfs.ext3*) автоматически делает резервные копии суперблока диска, просто на всякий случай.»

«Можно указать программе *e2fsck* проверять файловую систему, используя один из этих альтернативных суперблоков, при помощи опции **-b**, за которой следует указать номер блока этого альтернативного суперблока внутри проблемной файловой системы. Первый из альтернативных суперблоков обычно создается в блоке 8193, 16384, или 32768, в зависимости от размера вашего жесткого диска». Полагая, что диск большой, фон Хаген пытается использовать как альтернативный последний из этих блоков:

```
fsck.ext3 -b 32768 /dev/hda1
```







Сети от 5 до 20 машин

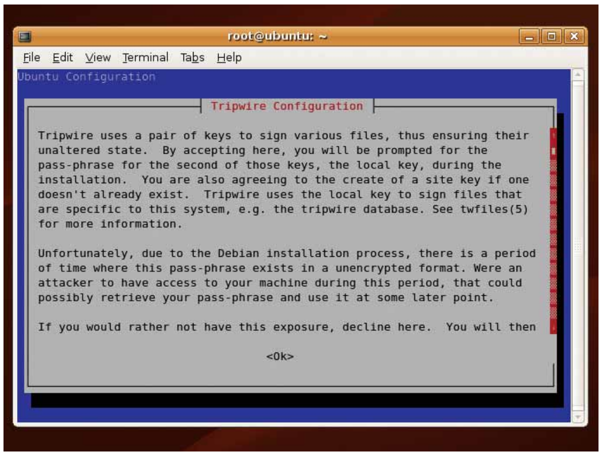
Безопасность: Установка Tripwire



Чтобы знать о каждой осуществленной атаке на вашу систему, попробуйте *Tripwire* [мина-растяжка – *англ.*], систему обнаружения вторжений. *Tripwire* хранит уникальный хэш для каждого системного файла, установленного на вашем сервере, и раз в день сверяет важные системные файлы со своими хэшами в базе данных. Если что-нибудь изменилось, хэш не совпадет, и *Tripwire* предупредит вас об этом.

Установка *Tripwire* очень проста, но очень важно сделать это в нужный момент. Незачем включать в первое сканирование сомнительные двоичные файлы! В идеале *Tripwire* устанавливается на свежую инсталляцию, содержащую только официальные заплатки безопасности для вашей системы. Установив пакет *Tripwire*, запустите простой настроечный скрипт.

Tripwire требует ключ узла для аутентификации сервера и локальный ключ для блокирования файлов. Затем следует инициализировать базу данных хэшей командой **tripwire --init** и ввести локальную парольную фразу. После этого проверьте целостность вашей системы командой **tripwire --check**. При сигнализации о проблеме в случае, когда вы доподлинно знаете, что ничего не менялось, наверное, следует подправить файл политики, **etc/tripwire/twpol.txt**.



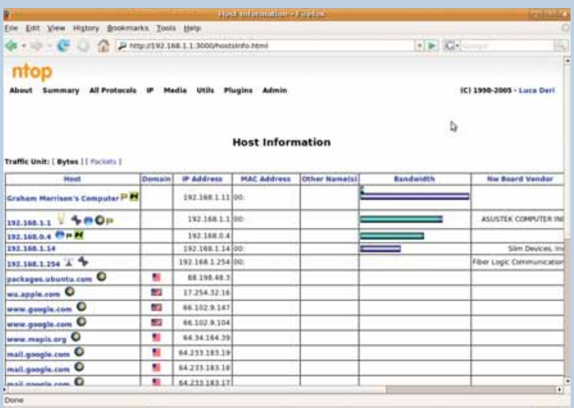
› ***Tripwire* имеет собственный скрипт настройки.**

Работа с сетью: Слежение за трафиком



Широкополосный доступ – дорогая штука. Подобно бухгалтеру, озирающему недреманным оком финансы компании, системный администратор должен точно знать, как этот ресурс расходуется вашей сетью. Утилита для данной работы называется *ntop*. Будучи запущенной, она анализирует каждый исходящий и входящий пакет данных, представляя результат в виде набора таблиц и круговых диаграмм на собственном настраиваемом web-сервере. Если *ntop* оставить работать на вашем сервере, вы сможете выстроить точную картину использования данных вашей сетью: самые занятые дни и времена суток, наиболее популярные соединения, а заодно и выяснить, не злоупотребляет ли кто-нибудь доступом.

Менеджер пакетов вашего дистрибутива должен поддерживать установку *ntop*. Если процесс не запускается автоматически, скомандуйте **/etc/inid.d/ntop start**. Просмотр данных на сервере, где работает *ntop*, осуществляется web-браузером через порт 3000. Нередко сервер имеет несколько сетевых адаптеров; тогда добавьте дополнительные сетевые интерфейсы, приписав **-i ath0** к команде **ntop**. Вместо **ath0** подставьте имя вашего второго сетевого интерфейса, или список имен, если их больше двух. Используйте то же имя для интерфейса, которое появляется, когда вы запускаете **ifconfig**.



› **Приятные графики и таблицы – и полезные данные!**

Производительность: Оптимизация Samba



Если в вашей системе совместно используются файлы через *Samba*, то вы, наверно, уже знаете, что это может стать узким местом для производительности. К счастью, существует множество способов ее тонкой настройки, они помогут вам передавать файлы пользователям быстрее.

Одна из главных проблем возникает, когда к вашему серверу подключается несколько машин с Windows. Это происходит из-за способа реализации сетевого протокола. При настройке *Samba* по умолчанию любая большая сеть будет испытывать проблемы, так как каждый клиент норовит оповестить о своей доступности всю сеть. Решение – убедиться, что вы используете только один *Samba*-сервер на каждую подсеть, и добавить в файл конфигурации *Samba* (обычно **/etc/smb/samba.conf**) этого сервера опцию **wins support = yes**.

Однако самый большой выигрыш в производительности может быть достигнут благодаря применению условных блокировок (opportunistic locking, oplocks). Это означает, что клиент может локально кэшировать файл для обработки, пока этот файл не понадобится другому клиенту. Если другой клиент затребует тот же файл, перед передачей файла по сети первый клиент должен синхронизовать свои изменения с сервером. Но бывают случаи, когда один файл, например, база данных, читается множеством клиентов, и выгоды не получается.

Тем не менее, включение условных блокировок способно существенно украсить жизнь, и его можно активизировать поресурсно. Неплохо, например, включить его для домашних каталогов ваших пользователей, поскольку некоторые файлы домашнего каталога часто интересны только его хозяину. Просто добавьте следующие строки либо в глобальный, либо в локальный раздел файла конфигурации *Samba* (**/etc/Samba/smb.conf**):

```
oplocks = true
kernel oplocks = true
level2 oplocks = true
```



Работа с сетью: Блокирование вредных хостов



Всегда имеются люди, пытающиеся влезть в ваш сервер, хотите вы этого или нет. Например, запустив SSH-сервер, вы обнаружите, что различные IP-адреса постоянно добиваются несанкционированного доступа к вашей системе. Но и вы не беспомощны: можно заблокировать любой диапазон IP-адресов, которым вы не намерены дозволять соединение с вашими машинами.

Оборотная сторона такого запрета – если вдруг когда-нибудь вам понадобится доступ с одного из таких адресов, у вас ничего не выйдет; но пока вы не заблокируете целый континент IP-адресов, маловероятно, что такое случится. Ключом к блокированию адресов служит файл конфигурации **/etc/hosts.deny**. Всем перечисленным там IP-адресам будет запрещено соединяться с вашим сервером. Просто скопируйте и вставьте проблемные адреса из файла **/var/log/auth.log**, где перечислены попытки входа в систему. Возможно также использовать доменные имена, равно как и сам домен, чтобы заблокировать все соединения из этого домена – метод полезен, если обнаружено множество попыток, исходящих от одного провайдера или учреждения.

Если вы предпочитаете разрешительный доступ с ограниченного круга машин, лучше заблокировать сразу все соединения в файле **hosts.deny**, добавив следующую строку:

```
ALL: ALL
```

а затем перечислить все разрешенные IP-адреса в файле **/etc/hosts.allow** – антиподе файла **hosts.deny**. Все указанные там адреса будут перекрывать установки файла запретов.

Если число попыток SSH-соединений столь огромно, что скопировать все нехорошие IP-адреса вручную не представляется возможным, можно установить пакет **denyhosts**. Этот демон просматривает ваш файл **auth.log** и молча подсчитывает число попыток соединения со всех IP-адресов, а при достижении определенного порога добавляет их в **hosts.deny**. Это – великолепный способ остановить атаки по словарю на пароли ваших пользователей, так как удаленное соединение на четвертой или пятой попытке будет заблокировано демоном *hostdeny*.



› **Заблокируйте удаленное соединение с любого Интернет-адреса или доменного имени, указанного в файле конфигурации /etc/hosts.deny.**

Безопасность: Удобное резервирование с BackupPC

У каждого есть в запасе история о потере драгоценных данных из-за сбоя жесткого диска. Но, похоже, неважно, были ли эти данные утраченной симфонией Бетховена или итогом тяжкой работы целого года: большинство из нас снова сползает к привычке создавать резервные копии лишь при крайней необходимости. Чтобы обойти эту проблему, резервирование ваших данных должно быть как можно проще, и существуют десятки утилит, пытающихся это осуществить. Хороший пример – Advanced Maryland Automatic Network Disk Archiver (Amanda), она легко справляется с компьютерной сетью умеренных размеров, хотя немного капризна в использовании.

Возможная альтернатива – *BackupPC* (http://backuppc.sourceforge.net), в особенности, если вы сумеете найти пакеты для вашего дистрибутива Linux. Его легко и установить, и использовать, а управлять им можно через его собственный web-интерфейс. После установки прежде всего надо изменить опцию в файле **config.pl**, который обычно располагается в каталоге **/etc/backuppc**. По умолчанию для передачи резервных копий он использует Samba, ради совместимости с Windows, однако для машин Linux гораздо лучше подходит утилита *rsync*. Найдите '$Conf{XferMethod}' и измените 'smb' на 'rsync'.

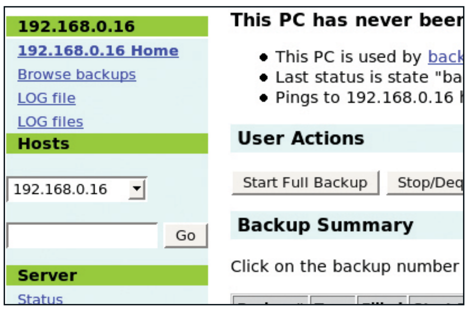
Важнейший момент в настройке *BackupPC* – обеспечение возможности SSH-входа пользователем 'backuppc' на сервере в учетную запись root на клиенте без необходимости вводить пароль. Скопируйте пару рабочих беспарольных ключей из двух других учетных записей в каталог **/var/lib/backuppc/.ssh** как на сервере, так и на клиенте – директории должны быть доступны для *BackupPC* (*о SSH без пароля см. стр. 43*). Остается только добавить IP-адрес каждого из ваших клиентов в файл **/etc/backuppc/hosts** и установить пакет **backuppc** на каждой из этих машин.

Наконец, запустите web-браузер и зайдите на web-интерфейс администратора для BackupPC на http://localhost/backuppc. Вам потребуется ввести в качестве имени пользователя 'backuppc', а пароль вам сообщался в процессе установки пакетов. Через этот web-интерфейс вы теперь можете соединяться со всеми клиентами, где запущен *BackupPC*, удаленно резервировать их данные, составлять график создания удаленных резервных копий и восстанавливать данные.



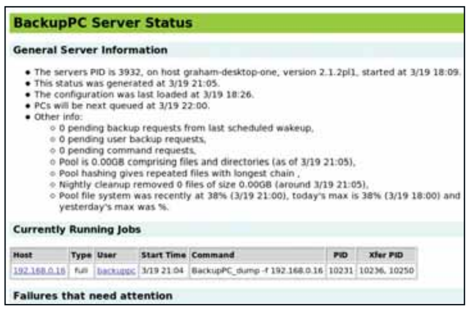
Поправка настроек Windows по умолчанию

В сети Linux вам следует чуть скорректировать файл конфигурации BackupPC и убедиться, что с любым клиентом можно соединиться без пароля.



Простая настройка

На сервере добавьте IP-адрес каждого клиента и щелкните на кнопке Start Full Backup. На машине-клиенте запустится процедура создания резервной копии.



Слежение за ходом выполнения

Вы можете отслеживать процесс создания резервной копии, и данные будут скопированы на сервер, откуда вы сможете восстановить файлы и каталоги.



Сети более чем из 20 машин

Производительность: Настройка пользовательских квот



Одна из наиболее важных задач системного администратора – равномерное распределение системных ресурсов по потребностям. Если одну и ту же машину у вас использует более десятка человек, то без активного контроля один пользователь может забрать львиную долю дискового пространства и процессорного времени, даже сам того не подозревая. К счастью, с помощью PAM наложить ограничения на учетные записи пользователей, исходя из их потребностей, сравнительно несложно. Как и в случае навязывания надежного пароля (стр. 42), повелителем пользователей оказывается снабженный нужным модулем PAM.

Модуль, управляющий квотами пользователей, называется Pam_limits. В отличие от модуля для надежных паролей, Pam_limits скорее всего установлен по умолчанию, но его все равно надо активизировать через соответствующий файл конфигурации PAM. PAM использует отдельный файл настройки для каждого поддерживаемого сервиса, и типичная установка включает файлы конфигурации для *Samba* и SSH. Для локальных пользователей, которые заходят в систему через менеджер входа Gnome, следует использовать файл конфигурации GDM. Через него, например, легко ограничить удаленных пользователей, одновременно оставив систему достаточно открытой для локальных. Просто добавьте в него следующее:

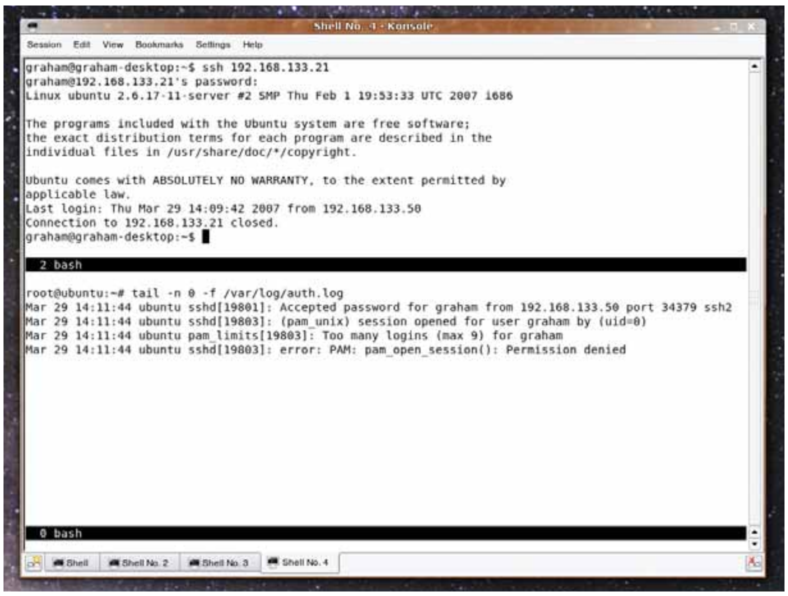
```
session required /lib/security/pam_limits.so
```

Ограничения как таковые задаются в файле **/etc/security/limits.conf**, используемом всеми сервисами аутентификации PAM. Формат этого файла следующий:

```
<домен> <тип> <величина> <значение>
```

В качестве домена может выступать имя пользователя или группы, а ‘тип’ (имеется в виду тип ограничения) может принимать значения «жесткий» (hard) или «мягкий» (soft). Разница между ними в том, что жесткий предел абсолютен и не обсуждается, а мягкий пытается пожаловаться в системный журнал; но доступ они зачастую ограничивают одинаково. Например, задав величину ‘maxlogins’ со значением ‘4’, вы жестко ограничите число одновременных входов четырьмя, независимо от типа ‘soft’ или ‘hard’.

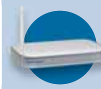
Ограничение числа одновременных входов – вполне разумная идея, особенно если у ваших пользователей только по одной учетной записи. Распознавание взломанной учетной записи становится куда проще, когда вы видите больше одного человека, пытающегося войти на сервер. Другие важные типы включают ‘priority’ для устранения помех, создаваемых процессами пользователя более приоритетным системным процессам; ‘fsize’, устанавливающий максимальный размер файла; и ‘cpu’, определяющий число минут, в течение которых пользователь может использовать CPU.



› В нижней панели разделенного окна вы видите запись журнала PAM о неудачной попытке входа, показанной в верхней панели.



Работа с сетью: Обновления



Большая ли, малая ли сеть у вас в распоряжении – время, потраченное на посещение каждой машины для установки обновлений, является временем простоя. Обновление всех машин Linux по всей сети под вашим контролем должно выполняться легко. Но, увы, не выполняется. Именно для этого и приходится содержать на предприятии Linux-хакеров, и если ваша организация достаточно велика для подобной роскоши, то это неплохая идея. Но и запуск ваших собственных обновлений по локальной сети также вполне возможен, при некотором планировании.

Лучшая из известных нам утилит для запуска обновлений – *SystemImager* (**http://wiki.systemimager.org**). Ограничение у нее одно, хотя и суровое; все машины, включенные в операцию обновления, должны иметь одинаковое аппаратное обеспечение, так как *SystemImager* поступает согласно своему названию: использует для каждой машины один и тот же инсталляционный образ, драйверы, приложения и прочее, и все машины, участвующие в обновлении, обязаны быть одинаковыми. Однако если вы обладаете преимуществом планирования заранее, то установить одинаковое оборудование в сети, где запущена одна и та же операционная система – разумная идея. Если ваша сеть достаточно быстрая, можно загружать каждую машину с одного образа, находящегося на сервере, используя лишь USB-носитель на клиенте и протокол BitTorrent для необременительной рассылки данных.

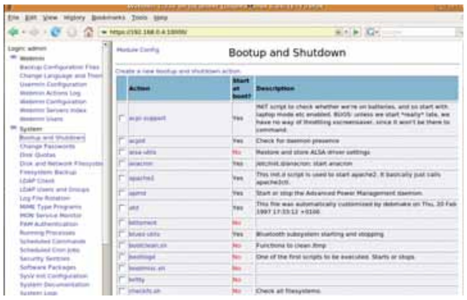
Имеются и другие способы обновления через сеть. Большинство дистрибутивов, включая Fedora, SUSE и Debian, позволяют автоматически скачивать и устанавливать любые версии обновлений. Например, Debian имеет утилиту под названием *cron-apt*, помогающую распланировать системные обновления посредством демона *cron*. Однако позволить системе обновлять себя автоматически – большая ошибка. Одно поврежденное или неправильное ядро, и всей вашей сети труба. Сперва надо обязательно протестировать обновление на паре машин, а уж потом запускать его в сеть.

Если тесты показали, что обновления безопасны (и оправдывают хлопоты), можете копировать их на остальные машины. Как? Установите локальный сервер обновлений для дистрибутивов, используемых на машинах-клиентах. Для упрощения этого процесса удалите приложения и службы, которые им не понадобятся – во избежание лишних движений. Для Debian можно развернуть локальный репозиторий. Просто поместите заготовленные файлы на сервер и настройте каждого клиента для использования вашего сервера в качестве единственного хранилища. В дистрибутивах Fedora и SUSE процесс похож: создайте локальное зеркало для репозитория обновления, содержащее только файлы, предназначенные для развертывания. В SUSE его URL настраивается через Yast, который синхронизирует каждого клиента с сервером.

Работа с сетью: Удаленное администрирование через Webmin

Утилит и опций конфигурации для озабоченного системного администратора на свете больше, чем байт в ядре, и они делают задачу управления различными серверами сложнее, чем она есть. Поможет *Webmin*: это интегрированная утилита удаленной настройки, из которой можно настроить в вашем сервере практически все. Она почти полностью заменяет SSH и командную строку. Даже если вам все-таки придется набирать команды, в *Webmin* предусмотрен SSH-клиент Java, и вы можете соединяться напрямую через свой браузер – отдельный SSH-клиент не потребуется.

Кое-какие требования, однако, имеются. Должен быть установлен Perl, вместе со своими модулями, и модуль аутентификации PAM для Perl. Понадобится также *OpenSSH*, для обеспечения безопасности соединения между вашим сервером и web-браузером. Затем скачайте последнюю версию *Webmin* с www.webmin.com и распакуйте архив куда-нибудь вроде **/opt/webmin**. Запустите скрипт *setup.sh* в новом каталоге и ответьте на пару вопросов, в том числе задайте пароль для учетной записи администратора. Откройте в браузере https://servername:10000, и вперед.



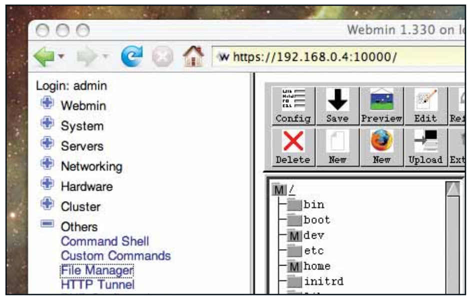
Хранилище настроек

Webmin освободит вас от запоминания каждой службы и файла настройки в собственной памяти.



Сложные задачи стали просты; ну, ладно – проще

Сложные настройки можно выполнить с помощью значков – хотя следует соображать, что вы делаете.



Доступ извне Linux

Работая с web-интерфейсом, вы получите доступ даже к SSH-клиенту и файловому менеджеру.

Безопасность: Инструменты для обратной связи с сервером



Иметь свежайшее ПО и контролируемое окружение сервера – еще далеко не все. Вам следует точно знать, что и когда происходит на машине, на которую полагаются десятки людей. Для этого, конечно, можно обратиться к системным журналам, содержащим все важное (а также много чего неважного), что произошло с вашей системой. Но никто ведь не проводит рабочее время за чтением этих файлов. Куда проще заставить что-нибудь изучить их за вас, и существует множество утилит для решения подобных задач!

Множество приложений разработаны для наблюдения за одним файлом журнала, например, создаваемым брандмауэром (*Fwlogwatch*), прокси-сервером *Squid* (*Squidalyzer*) или web-сервером *Apache* (*Analog*). Но можно также использовать одну из специальных утилит, способных одновременно отслеживать несколько файлов – что и требуется для сервера более общей конфигурации.

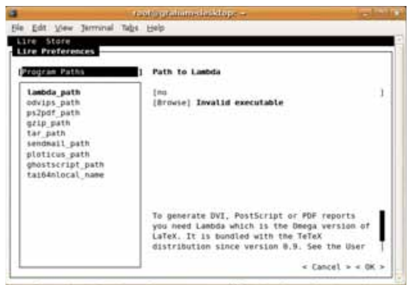
Популярна программа *Sentry Tools*, с уклоном на слежение за портами и соединениями с машиной, но мы остановимся на утилите мониторинга *Lire*. *Lire* доступна как через менеджер пакетов вашего дистрибутива, так и на сайте разработчика (http://logreport.org). *Lire* способна в одиночку уследить за серверами *MySQL* и DNS, электронной почтой (включая *Postfix* и *Sendmail*), брандмауэром *iptables*, FTP и системным журналом, наряду со многими другими. Она также умеет выводить отчет в виде текста, PDF или HTML – что делает ее незаменимой для ежемесячных отчетов.

Установив *Lire*, вам следует ее настроить. Ввод **Lire** в командной строке откроет простенький пользовательский интерфейс по типу curses; нажмите Control-x для вызова меню, затем спуститесь курсором до опции Preferences. Отсюда вы можете настраивать генерацию отчетов; здесь же *Lire* найдет утилиты командной строки, необходимые ей для работы. Если вы устанавливали *Lire* через менеджер пакетов, все основные опции должны быть преднастроены для вашего дистрибутива.

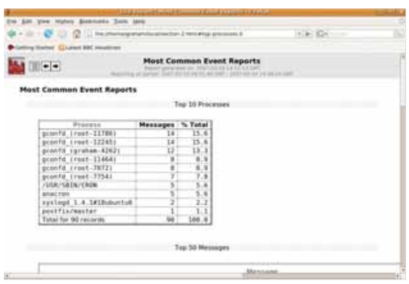
Чтобы разузнать, какие генераторы отчетов установлены и могут использоваться, запустите **lr_log2report --help dlf_converters**. Выведется список включенных генераторов, всех и не перечислить. Хорошим местом для старта будет **syslog**, так как это файл журнала, используемый для записи общесистемных событий. Создайте каталог для хранения созданного отчета (назовем ее **local**); для генерации HTML-отчета на базе syslog, введите

```
lr_log2report syslog /var/log/syslog --output-format html local
```

Можете также выбрать в качестве формата вывода **txt** или **pdf**, а если установлен пакет **ploticus**, то генератор включит диаграммы по данным отчета. Если вы довольны выводом, задайте для *Lire* расписание автоматического создания отчетов, с возможностью послать их вам по электронной почте через *Postfix*.



› Простенький интерфейс настройки *Lire* может немного покапризничать …



› …но отчеты, создаваемые ею, окупают затраченные усилия.

Идем дальше

При использовании Linux, вы всегда будете в курсе новых настроек и обновлений; в том-то и состоит системное администрирование. Если наши советы раздразнили ваш аппетит администратора, почему бы не рассказать нам об этом? Наши форумы всегда полны сплетен от сисадминов, и вы убедитесь, что они – отличное подспорье, если хочется побольше узнать о последних разработках, равно как и место, куда можно обратиться за помощью и за советом. www.linuxforum.ru LXF



О СОТРУДНИЧЕСТВЕ

«Главное — учиться друг у друга и применять научный подход для улучшения любой программы.»

Фото: Jena Cumbo



Сохраняя веру

Уход из Novell сделал его героем мира Open Source, но Джереми Эллисон просто хочет работать над *Samba*. И играть в *Half-Life*.

Мир Linux рассуждал о важности союза Novell и Microsoft, когда в декабре прошлого года всех ошарашила новость о том, что Джереми Эллисон [Jeremy Allison], один из создателей *Samba* и сотрудник Novell, покинул фирму в знак протеста против этой «ошибки», которая, по его мнению, «повредит успеху Novell в будущем». Было понятно, что этот союз сильно отразится на Linux.

Эллисон, чья принципиальная позиция вызвала общие аплодисменты обычно конфликтующих фракций компьютерной сферы, обнаружил, что совершенно не в состоянии совместить патентное соглашение Novell с идеей свободного ПО. «Мое основное убеждение насчет этой сделки – я полагаю, что даже если буква лицензии и не нарушена, то нарушен дух лицензии GPL (под которой выпускается код *Samba*), предписывающий одинаково относиться ко всем получателям кода», – сообщил он тогда в своем уведомлении об уходе, разосланном электронной почтой. Его возражения от имени *Samba* подняли кое-какие интересные вопросы о существующей напряженности между фирмой и сообществом и о том, как хакеры могут сочетать вклад в свободное ПО с оплачиваемой ежедневной работой, даже если, как в случае с Эллисоном, эта ежедневная работа включает в себя такую интересную и ценную деятельность, как корпоративное внедрение SUSE Enterprise Linux.

Как сообщалось, Эллисон ушел в Google, где ему дали полную свободу работать над *Samba*, технологией сетевого взаимодействия. **Ник Вейч** встретился с ним на конференции LinuxWorld в Нью-Йорке.

Linux Format: **Ну, с чего начнем?**
Джереми Эллисон: [Смеется] Не знаю.

LXF: **Давайте начнем с больной темы – с я-больше-не-работаю-в-Novell. Вы считаете, что можно заниматься коммерцией в пространстве Linux, избегая конфликта интересов?**
ДжЭ: Конечно. Я надеюсь, что как только выйдет GPL 3, и некоторые проекты перейдут на нее, всё изменится, и Novell... Короче, все опять будут счастливы.

LXF: **Вы думаете, Novell будет счастлива принять на GPL 3?**
ДжЭ: Да. Насколько я понимаю, GPL v3 никак не повредит бизнесу Novell. Не забудьте, что я говорю только о том, что касается *Samba*.

LXF: **Полагаю, GPL 3 сможет многое прояснить.**
ДжЭ: Думаю, да. Полагаю, нам именно это и надо.

»

» **LXF: Кажется, часть критикующих GPL 3 делает это потому, что кроме юридических положений, в ней содержится и ряд политических.**

Дж.А: Но так всегда было в GPL. Вспомните манифест GPL Manifesto в исходной лицензии. Это – политическое заявление, и так всегда и было.

LXF: Она не может не быть политической. Но это, по-моему, кое-кого пугает. Они рассуждают так: «Я что, действительно должен это принять?»

ДжЭ: Я смотрю на это так. Я был на конференции в Лондоне [LinuxWorld] – по-моему, это было в октябре – и один юрист там заявил: «О, GPL3 – это тот самый ужасный *Коммунистический Манифест* со злостными перегибами! По сравнению с чудесной, общепонятной GPL 2 он отвратителен». Я просто расхохотался. За последние 15 лет я посидел и переговорил со множеством корпоративных юристов, и все они твердили: «GPL 2 – это же коммунизм, это ужасно!» Я и говорю ему: «Как знать, может быть, лет через десять вы скажете, что GPL 4 ужасна – в отличие от разумной и умеренной GPL 3!»

LXF: Как-то я спорил с одним человеком о компьютерном оборудовании. Он жаловался, что Linux сложен и ужасен в использовании и что люди его не понимают. Он сам пробовал, но не добился работы ни с каким его оборудованием.

ДжЭ: Что, это был Red Hat 5.1?

LXF: Это был Ubuntu – по-моему, одна из последних версий. Но он, например, сказал: «С моим iPod это работать не будет» – а я ему ответил, что можно установить разные программы, которые будут успешно обмениваться информацией с iPod.

А он мне: «Ну, да, но музыку, которая у меня есть, они не воспроизведут, потому что она вся в AAC» или в каком-то там еще дурацком DRM-ном формате. Я не смог его убедить, что это не Linux не работает, а Apple «подсадила его на свою иглу».

ДжЭ: Значит, для него Linux не работает. У него есть все основные приложения, которые ему нужны. Многим людям по той же причине нужна Windows. Если есть приложение, которое делает то, чего нет в Linux или еще что-то в том же роде, я совершенно спокойно говорю: «OK, пользуйтесь Windows». Я не хочу заставлять людей заниматься ерундой. Кроме моей мамы – она мой испытательный полигон!

LXF: Просто раздражает иногда, что некоторые считают Linux ерундой, раз ничто на свете не заставит его воспроизвести DRM-ные файлы из iTunes.

ДжЭ: Возвращаясь к Novell: по-моему, недостаточно громко прозвучало, что Peugeot установил с помощью Novell 20 000 настольных Linux-систем. Нельзя говорить, что Linux не готов для использования в рабочих станциях, если Peugeot ставит таких 20 000 штук. Я работал в этом проекте перед увольнением, хотя моя роль в нем была весьма скромной. И мы говорим не о каких-то там настольных машинах, которые будут использоваться графическими дизайнерами или еще кем-то, кто собирается применять инструменты Linux, а о владельцах ноутбуков, которые таскают их домой, работают вне сети, в отелях, через VPN – и у них все это получается.



LXF: Естественно, Novell уже использует Linux на своих рабочих столах.

ДжЭ: Если хотите знать о настольном Linux, то, если честно, вам надо говорить с другими людьми. Или с моей мамой.

LXF: И каковы ее предпочтения в Linux на данный момент?

ДжЭ: На данный момент она исследует SUSE 10.1, но, возможно, я ее переведу на Ubuntu. Я подумываю о том, чтобы подарить ей один из этих Mini PC.

LXF: Давайте немного поговорим о Samba: как там дела с Samba?

ДжЭ: Много разработок. И эти разработки все разрастаются и разрастаются. Сейчас имеется четыре различных CIFS-клиента, независимых от Windows. Есть CIFS Стивена Френча [Steven French] в Linux, есть клиент в Mac OS X; не знаю, могу ли я назвать еще одного… давайте назовем его A.N. Другой крупный разработчик Unix взял код Darwin и создает еще одного CIFS-клиента, и мне очень хочется помочь им в этом. Затем, естественно, у HP имеется свой клиент на базе Sharity.

Так что есть четыре разных CIFS-клиента, и все они реализуют расширения Unix. В этом пункте возникает интерес у тех, кто работает с NetApp и EMC, и они уже могут сказать: «Если вы напишете все эти расширения на RFC, мы, возможно, включим их в свой код». И тогда у вас получается альтернативная версия протокола CIFS для настольной системы. Потому что, по моей теории, все будут использовать CIFS на своих столах. NFS v4 или прочие протоколы файлового доступа на заднем плане – это хорошо. Но на настольной системе всем приходится держать Windows, так что они уже работают с CIFS, а чтобы другие системы с ней хорошо взаимодействовали, им тоже придется работать с CIFS. Но у вас может быть разновидность CIFS, созданная под POSIX, и в таком случае вы обязательно расширите этот протокол, чтобы он был более полезен для вас. Наличие для этого четырех независимых клиентов означает наличие весьма гибкой кодовой базы, которая выявляет большое количество ошибок, деталей применения, и в принципе создает сильную экосистему. Да еще есть всякие разновидности Windows, которые так и так приходится поддерживать.

> **О КОЛЛЕГАХ ПО SAMBA**
>
> **«Я нахожусь там, где полно людей, которые говорят: «Ты – идиот, все надо делать не так!»**

Вот так обстоят дела, еще есть работа с операцией отсоединения Winbind, которую впервые применили в Novell и которая использовалась при развертывании в Европе, это очень крутая штука. Я там больше не работаю, но штука действительно крутая.

Еще есть кластерная *Samba*, над которой работает Тридж [Эндрю Триджелл – Andrew Tridgell] [CTDB], так что я усвоил неплохие навыки кодирования.

LXF: Наконец-то!

ДжЭ: Да уж, наконец-то! Я нахожусь в таком месте, где полно людей, и каждый мне говорит: «Ты – идиот, все надо делать не так!» А я им в ответ: «А знаете, вы правы!» Мы занимаемся серьезной реструктуризацией внутренней структуры *Samba*. Вроде чистки Авгиевых конюшен: по одному стойлу за раз. Мы вычищаем старый код. Это как будто вы летите на одном самолете, и при этом строите другой самолет – переносите части из одного в другой на лету. И это здорово.

LXF: И, очевидно, вам это тоже нравится.

ДжЭ: Да. Мне нравится создавать код.

LXF: И давно ли вы этим занимаетесь?

ДжЭ: Думаю, лет 12, может быть, 15. Давно уже.

LXF: И как вам удается сохранить мотивацию?

ДжЭ: Потому что здесь все постоянно меняется. Это как с водопроводчиками: некоторым нравится быть водопроводчиками, потому что им интересно работать с трубами, и я один из таких. Ты просто наводишь глянец и думаешь: «Да, уже блестит, но можно сделать так, чтобы блестело еще ярче. Можно стукнуть по этой трубе, и она станет поровнее». Не знаю. Почему Линусу нравится писать ядро? Почему парням из *Apache* нравится заниматься *Apache*? Кого-то по дороге теряешь, кого-то находишь, но это проект, и работать над ним – удовольствие; до сих пор удовольствие. Круто, когда приходишь во Fry, и там все эти ящики, и ты знаешь,

что все они работают на твоем коде. А потом покупаешь один из них, смотришь, и начинается: «О, Господи, они работают на 3.02! Там же столько ошибок!» Недавно ко мне обратился один разработчик NAS-систем, потому что вышла Vista, и у них все сломалось. Я ему говорю: «Почему это вышло из строя? Мы и тут тестировали, и там тестировали!» А потом оказалось, что они работали в режиме коллективной безопасности. «Ребята, вы что, сопроводиловку не читали? Мы его пометили как устаревший пять лет назад!»

LXF: Раньше никто не жаловался.

ДжЭ: Да, верно, потому мы и работаем над этим. Это здорово, когда много работы.

LXF: У вас никогда не возникало желания заняться чем-то еще?

ДжЭ: Помню, я на какое-то время ушел в мир Kerberos – это было в 95–96 году – и занимался еще и *Cygwin*. Если бы я только и занимался файлово-серверным кодом, то, конечно, мне бы это надоело. Но *Samba* сейчас включает солидные куски Kerberos, LDAP, динамического обновления DNS, шифрования, расширения POSIX CIFS, помощь в создании ядра Linux, помощь парням из Mac с их программами, поддержка потоков… Как там эта японская игра называется – *Катамари Дамаси*, или что-то в этом роде – надо катать большой мяч, а цель игры – чтобы как можно больше вещей к вам прилипло? Вот и у нас то же самое.

LXF: Это почти что готовая ОС.

ДжЭ: Я как-то пошутил, что нам надо в какой-то момент переименовать SMBD в NTOSKRNL.EXE, и будет нормально. Конечно, это не слишком политкорректно, но… *Winbind* ведет нас к замене интерфейсов NSS (Name Server Switch), потому что, не знаю, заметили вы это или нет, но NSS – это отстой. В этой области Windows действительно в своем праве, и она намного, намного опережает Linux и Unix. Что хорошо в Windows – так это то, что они внедряют исключительно крупные службы каталогов по всему миру. Посмотрите на современный интерфейс Windows: вы хотите добавить ACL к файлу, так? Раньше, в *NT4* и более ранних версиях, надо было вывести диалоговое окно, и там было бы написано, сами знаете – List Users, верно? Элемента List Users больше нет. А причина в том, что перечисление абсолютно всех пользователей, к которым вы можете обратиться, может означать, что вам придется обратиться к DC (контроллерам доменов) в Европе, Бангалоре, Японии – это ведь просто невозможно. На это уйдут часы. Половина из них может быть вне доступа, понимаете?

А сейчас у вас есть поисковый интерфейс. Если вам надо добавить пользователя, вы вводите поисковый запрос. Это именно то, что надо. На данный момент мы ограничены интерфейсами POSIX/NSS, которые работают так: «начать перечисление пользователей, следующий, следующий». Здесь нет, например, «найти всех пользователей, начинающихся с V».

LXF: В некотором смысле по-прежнему выгодно смотреть за тем, что используют в Windows, и учиться на их ошибках?

ДжЭ: Да, безусловно. И, надеюсь, они тоже учатся у нас. Самое главное, по крайней мере, для меня, учиться друг у друга и использовать научные методы для улучшения программ. Для меня суть Open Source именно такова: это научный подход, приложенный к созданию программ. Вы добиваетесь результата, не просто рассматривая то, над чем приходится работать; вам приходится рассматривать множество других систем, чтобы узнать, что в них хорошего и чему вы можете у них научиться.

LXF: Каких волнующих событий вы ожидаете в этом году? Что-нибудь интересное произойдет?

ДжЭ: Вы имеете в виду, в *Samba*? Там тонны работы. Просто тонны. Я хочу упростить удаленный доступ, чтобы можно было развернуть серверы *Samba* действительно легко. Я надеюсь, что этот год станет годом, когда дистрибутивы всерьез займутся взаимодействием с Windows. Novell уже продемонстрировал, как это делать, так что я надеюсь, Ubuntu и Red Hat заявят: «Эй, нам это тоже надо!»

LXF: Думаю, в Ubuntu над этим уже работают.

ДжЭ: Если к концу года добьются уверенного взаимодействия с Windows – через



Active Directory или что-то другое – и оно будет доступно всем, каждый сможет работать на ноутбуке, отсоединенном от сети, для нас это и будет волнующим событием.

О ДИСТРИБУТИВАХ

«Волнующим событием для нас стало бы уверенное взаимодействие с Windows к концу года.»

LXF: Это достижимо?

ДжЭ: Да. В особенности потому, что не мне придется выполнять эту работу! Код уже есть, его нужно адаптировать, его нужно ввести в дистрибутивы, и нужно, чтобы были соответствующие интерфейсы управления. И вот над этим-то нам и приходится работать, и это ужасно для нас. Мне нравится Ubuntu: он очень хороший, очень милый. Это – один из тех дистрибутивов, в которых все отработано, и все очень хорошо. Я купил фотопринтер HP, и он сразу заработал; можно подключить камеру, и я импортировал все, что надо, в *F-Spot*, подключил кинокамеру и скопировал MPEG, и потом записал DVD – все отлично работает.

LXF: Даже под Windows с этим справиться нелегко.

ДжЭ: Знаю. Проблема этих приложений в том, что они отличаются от тех, с которыми работают пользователи Windows, а люди не любят изменений. И с играми отстой. Играть нельзя, ну, не знаю, какими бы ни были игры. Я вот люблю *Half-Life*, но…

LXF: Она малость устарела!

ДжЭ: Ну, *Half-Life 2*. Я приложу максимум усилий, чтобы она запустилась в Wine! Наверное, именно это отнимает у меня большую часть времени, которое я мог бы посвятить *Samba*.

LXF: Уверен, что быстрая стрельба из больших пушек вдохновляет вас на великие идеи…

ДжЭ: Да нет, (*рычит*) просто люблю убивать.

Читайте больше!

Наше полное интервью с Джереми – на www.linuxformat.co.uk/mag/allison.html. Предупреждение: в процессе обсуждаются комиксы.



Музыкальная Linux-студия

Приключения ЗВУКА

Готовы к новому проекту? **Амиас Ченнер**, энтузиаст электронной музыки и ветеран Linux, расскажет, как легко построить музыкальную студию на открытом ПО.

Любой, кто хоть немного занимался электронной музыкой или бывал в студии звукозаписи, вспомнит этот момент: «Ну вот, опять зависла». У Linux-пользователя подобные фразы всегда вызывают улыбку, и я невольно задумываюсь о том, как повел бы себя в такой ситуации Linux. И вот, когда моя студийная система на Windows в очередной раз «поплыла», я решил поискать выход: заменить аудио-программы Windows открытыми эквивалентами и построить музыкальную Linux-студию – а затем испытать ее с каким-нибудь новым хитом.

Раньше со звуком на Linux работали только одержимые, но заплатки ядра для поддержки приоритетов реального времени и растущее качество дистрибутивов сделали идею реально привлекательной – в особенности при неприятной необходимости поддерживать коммерческую ОС. Все больше музыкантов предпочитают программы инструментам, и все больше музыкантов обзаводятся собственными студиями. Качество профессиональных инструментов просто не поспевает за качеством ПО.



› **Главный экран *Ardour* с несколькими дорожками записей.**

Существуют умелые, талантливо написанные и надежные коммерческие аудиопрограммы, но для многих полупрофессиональных пользователей они слишком дороги, чересчур требовательны к аппаратным ресурсам или сложны в изучении, чтобы соблюдать соотношение цена/качество, особенно если совместимость или надежность не гарантируются. Иногда единственный способ избежать начальных расходов – установка нестабильной пиратской версии, но он требует продолжительного освоения и встраивания программы в креативную аудиосреду. Для увлеченного музыканта прервать полет вдохновения из-за «жучков» – последнее дело, поэтому качество ПО очень важно. На Linux можно, как минимум, «отремонтировать» ПО – и, естественно, не нужно за него платить.

Сцена открытого ПО

Ранние версии *Rosegarden*, MIDI-секвенсора и редактора, и *Ardour*, многоканального рекордера и редактора, имели пугающе примитивные интерфейсы. Аудиопрограммы для Linux традиционно использовали стандартные органы управления, в основном негодные или, в лучшем случае, неудобные. Можно считать, что настраиваемые органы управления – не главное в аудиопрограмме, но, сделанные с умом, они способны создать очень удобную и отзывчивую рабочую среду.

Качество интерфейсов, впрочем, улучшалось, и хотя FOSS пока не получил что-то столь же впечатляющее, как проприетарные *Ableton*, *Nuendo* или *Reason*, разрыв стремительно сокращается. В изобилии распространяя общедоступные, загружаемые CD, содержащие полные комплекты ПО на все случаи жизни, Linux может сделать электронную музыку доступной повсеместно, наравне с акустическими инструментами.

Часть 1 Настройка

Как я настроил выбранное ПО Linux

Как минимум, понадобятся многоканальная звуковая карта, приличный микшер и хороший набор колонок. В моем случае один очень щедрый мой друг одолжил мне большую часть профессиональной оснастки, которой он пользовался для звукозаписи в студии на базе Windows. Конфигурация моей машины: Pentium 4.2 ГГц (1 ГБ ОЗУ и 120 ГБдиск UDMA). Прежде чем настраивать Linux для работы с аппаратурой, нужно было принять важные решения в отношении ПО: ведь взяв негодные инструменты, можно загубить творчество бесконечными заплатками, конверсиями и – о ужас! – математикой.

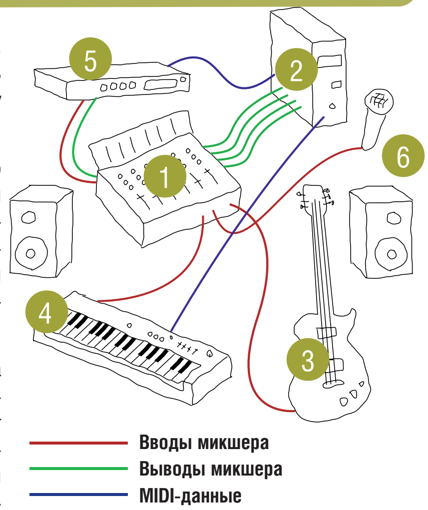
Первый вопрос – выбор дистрибутива. Хотелось найти дистрибутив, заточенный под аудио, снабженный музыкальным ПО и допускающий несложное добавление программ. Основными претендентами были 64 Studio, Studio To Go, Agnula и Ubuntu Studio. В конце концов я остановился на 64 Studio, так как это небольшой дистрибутив на основе Debian, включающий все стандартные аудио-инструменты Linux, настроенные на работу с Jack в реальном времени. Кроме того, есть преднастроенная поддержка кластеризации и SMP. Пусть мой компьютер не 64-битный и не многопроцессорный, но лучше вырабатывать стандарт на дистрибутиве, нацеленном на будущее. Было также желание обойтись без компиляции.

Подбор ПО не составил труда, ведь в 64 Studio для выполнения любой работы предназначена единственная избранная программа. Все программы скомпилированы с поддержкой 64-битной SMP, 96-КГц аудио и совместимыми версиями ПО для любых подключений. Со многими коммерческими программами использование SMP требует дополнительной лицензии.

Сборка

Итак, что для нас приготовил 64 Studio? *Ardour* отвечает за многоканальную запись, предоставляя несколько аудио-вводов, выводов и каналов, с помощью которых можно использовать LADSPA или VST для вставки эффектов. Audacity будет аудио-редактором, альтернативы ему просто нет. *Jamin* предоставляет мощный инструмент тонкой настройки миксов с 1,024-полосным эквалайзером и компрессором. MIDI-секвенсор – *Rosegarden*, это очевидно. Он работает с MIDI- и аудио-модулями (LADSPA, DSSI, а скоро и VST), но не будет воспроизводить аудио-файлы, если не установить соответствующий модуль. Для ударных есть *Hydrogen*, обладающий мощным аранжировщиком барабанных паттернов, базовыми возможностями управления сэмплами и управлением MIDI, которое плавно интегрируется в *Rosegarden* через процедуру импорта файлов.

Linux-студия звукозаписи



1 Микширующая консоль Это центр студии, все аудиовходы проходят сквозь консоль перед передачей на Linux-машину по одному из четырех отдельных выходных каналов.

2 Linux-компьютер При четырех входах можно одновременно записывать с микширующей консоли два стереоканала. Большее количество входов позволяет записать более широкий диапазон. Отсюда также отправляется и принимается MIDI для секвенсирования синтезатора.

3 Электрогитара Требует особого аудио-входа под названием high-Z, для адаптации гитарного аудиосигнала. Таким вводом располагают очень немногие звуковые карты, что вынуждает использовать микширующую консоль или вывод от специального устройства для генерации гитарных эффектов.

4 Клавиатура синтезатора Хотя клавиатура не производит звук, она часто используется для отправки MIDI-нотных данных на Linux-машину. Если необходимо воспроизведение звука, то аудиовывод должен быть подключен либо к микшеру, либо к компьютеру.

5 Внешние эффекты Воспроизведение некоторых эффектов, в особенности реверберации, лучше предоставить внешним устройствам, чем программе. Используя канал отправки эффекта на микшер, можно отправить некоторое количество звука и смешать возвращенный аудиосигнал с первоначальным фрагментом.

6 Микрофон Настоящему микрофону необходимо 48-В питание, подведенное от предусилителя через особый трехштырьковый XLR-разъем. Большинство микшеров оснащается несколькими предусилителями, но в комплект звуковых карт они включаются редко.

Инсталлятор 64 Studio, совершенно стандартная текстовая штука, за пять минут провел меня через процесс разбивки диска и создания пользовательской учетной записи. Я выбрал единственный раздел, но для коммерческой студии было бы разумнее применить несколько учетных записей и дисковые квоты. Создание отдельного раздела для журнала поможет предотвратить захват системы единственным пользователем.

При выборе звуковой карты для профессионального использования выгоднее быть разборчивым: у карт потребительского уровня (на вас смотрю, Creative Labs) скверное соотношение сигнал/шум, способное привести к появлению в ваших миксах совершенно неожиданных пассажей. Включив звуковую карту потребительского уровня в студийный микшер полуприличного качества, вы будете неприятно удивлены тем, сколько треска и хруста производят движения мыши и обращения к винчестеру. При использовании мощной звуковой аппаратуры подобные шумы могут лишить вас слушателей.



Я воспользовался звуковой картой Echo Layla3G 8-in/8-out PCI с коммутационным боксом. Кроме Echo Digital, Linux поддерживает карты RME, Event и Edirol – все это старые брэнды, поддержка в Linux новых карт пока недостаточна. Это огорчает, но старые карты все же очень широко распространены. К тому же есть причины для оптимизма в свете недавнего заявления Грега Кроа-Хартмана [Greg Kroah-Hartman], обращенного к производителям, о поддержке »

ПО для студии

В Linux имеется ПО для всех видов студийных работ. Чтобы помочь вам сориентироваться, здесь приведены Linux-эквиваленты известных программ.

ПО	Windows	Mac	Linux
Звуковой редактор	Adobe Audition, Audacity, Sound Forge, Wave Lab	Audacity, Sound Studio	Audacity
Многоканальная запись	Adobe Audition	Ardour, Digital Performer, GarageBand, Logic Audio	Ardour
MIDI-секвенсор	Cakewalk, Cubase	Cubase, Logic Audio, Metro	Rosegarden
Ударные	FruityLoops, Reason	Doggiebox, iDrum,	Hydrogen
Программные синтезаторы	Ableton, FruityLoops, Reason	Ableton, Reason, VirSyn	QSynth, ZynAddSubFX



» разработки Linux-драйверов для их устройств (www.kroah.com/log/linux/free_drivers.html). 4FrontTechnologies (www.opensound.com) тоже разрабатывает драйверы для популярных карт профессионального уровня от M-Audio: линеек Audiophile и Delta.

Обычно профессиональные звуковые карты имеют надежно изолированные чипы, хорошо заземленные и экранированные от электромагнитного шума, производимого окружающей аппаратурой. Карты USB и FireWire реже страдают от шумовых проблем, так как находятся обычно за пределами системного блока. К счастью, Linux-поддержка USB- и FireWire-карт значительно улучшена благодаря надежной работе с общими классами аудиоустройств и выпуску Freebob, драйвера для ALSA и Jack.

Скрипты реального времени

Два скрипта для просмотра и настройки приоритетов IRQ реального времени:

check_irq_priority

```
#!/bin/bash
chrt -p `pidof "IRQ $1"`
```

set_irq_ priority

```
#!/bin/bash
chrt -f -p $2 `pidof "IRQ $1"`
```

Низкая латентность

В 64 Studio я сразу направился в *QjackCtl* (интересное название для альбома, а?), чтобы узнать, правильно ли настроена моя карта Layla3G с Jack. Увы, нет. Зато работала ALSA. Все, что мне оставалось сделать – это изменить количество входных и выходных каналов на восемь, установить частоту дискретизации 96 КГц и задать возможно более низкую латентность. С тех пор система не знала простоя и действовала с такой производительностью, что регулярно переполняла винчестер.

> «Linux-поддержка USB- и FireWire-карт значительно улучшена.»

Затем я загрузил *Ardour*. Хотя все было готово к немедленной записи, первые два канала оказались не в состоянии взяться за дело. Стандартная конфигурация ALSA на 64 Studio не затрагивала микшер, это осложнило диагностику проблемы. Карту Layla3G лучше всего настраивать с помощью программы *Echomixer*, позволяющей связывать между собой любые входные и выходные порты, а также настраивать уровни усиления и подавления. Но это не спасло, и мне пришлось заключить, что два первых канала на Layla3G настроены на цифровой ввод и вывод вместо аналогового.

Управление аудио с помощью Jack и эффективно, и просто в изучении. Jack предоставляет широкий выбор вариантов связывания входов и выходов в стиле «многие со многими». Выводами могут считаться не только выводы звуковой карты, но и вводы аудиопрограмм; вводы могут быть не только вводами звуковой карты, но и выводами аудиопрограмм. Можно ненароком создать обратные петли, так что будьте осмотрительны. *QjackCtl* упрощает управление этими связями с помощью специальной соединительной кнопки.

Советы по звукозаписи

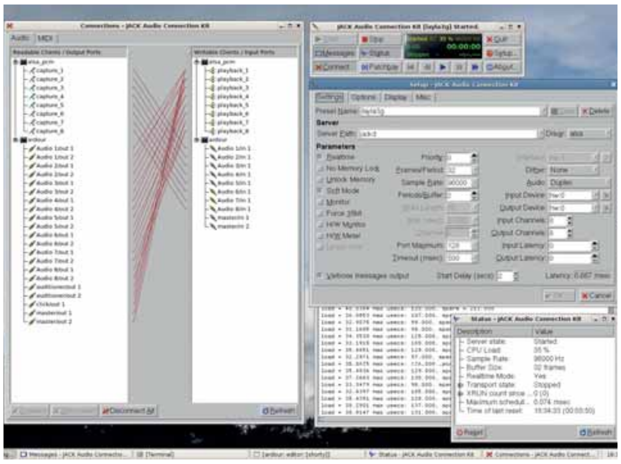
» Чтобы упростить работу с *Rosegarden*, установите псевдонимы для ваших устройств в MIDI-менеджере. Некоторые распространенные устройства обладают даже банками профилей, с корректными параметрами и названиями патчей. Загрузить их можно, нажав кнопку Banks (Банки) в MIDI-менеджере.

» Получив работоспособную конфигурацию студии в *Rosegarden*, можно сохранить ее как «студию по умолчанию». В этом случае уже при загрузке *Rosegarden* все инструменты будут настроены автоматически. Перейдите к Composition > Studio > Save Current Document As Default Studio (Композиция > Студия > Сохранить текущий документ как студию по умолчанию).

» Многим нравится запускать Jack одновременно с началом сессии *gnome-session*. Чтобы добиться этого, просто выберите пункт Sessions (Сессии) в меню Desktop > Preferences (Десктоп > Параметры) и добавьте *QjackCtl* в список автозагрузки. Если вы хотите, чтобы автоматически запускался транаспорт Jack, воспользуйтесь вместо этого *qjackctl -s*.

» Сохранение маршрутизации Jack для последующего использования предотвратит трудоемкую перезагрузку студийной конфигурации для каждой дорожки. Просто щелкните Save (Сохранить) в разделе наборного поля *QjackCtl*.

» Отображение маршрутизации Jack иногда сбивается. В этом случае щелкните New (Создать) для обновления раскладки с использованием данных о текущем состоянии подключений.



› **Сочиняйте прекрасную музыку с помощью аудио-интерфейса Jack.**

Проблема Motu

Единственным устройством, не поддавшимся настройке, было Motu MIDI Express 128 – к моей досаде, так как оно управляет MIDI-синтезаторами. Я посоветовал бы избегать аппаратуры Motu: лучше пользоваться устройством от M-Audio или дешевым MIDI-интерфейсом USB.

Я попробовал три синтезатора: станковый Access Virus, установку Novation Super Bass и Roland Juno-60 с интерфейсом Kenton Pro-DCB MIDI, и нашел, что управлять ими можно, связав MIDI-выводы от Layla3G через сквозные порты MIDI на синтезаторах. Иметь специальный MIDI-порт для каждого синтезатора было бы неплохо, но это пока подождет. Моя MIDI-клавиатура Evolution MK-249C работала превосходно.

Оптимизация в реальном времени

Упомянутые мной заплатки ядра реального времени позволяют процессам работать в условиях приоритета жесткого реального времени, игнорируя прерывания, имеющие более низкий приоритет. В случае с аудио можно присвоить наивысшие приоритеты звуковой карте, за ней жесткому диску – тогда получится непрерывная аудиозапись. Это достигается передачей *chrt* (часть пакета *schedutils*) ID процесса поточной версии прерывания для каждого устройства, а также уровня приоритета для каждого процесса. Чтобы просто посмотреть уровни приоритета, можно ничего не вводить.

Приоритеты реального времени – это, конечно, риск для большинства систем, поэтому управлять ими может только администратор. Но существуют механизмы для передачи его полномочий избранным пользователям. *Realtime-lsm* – наиболее популярный из них, но в настоящее время эта программа почти полностью уступила место *Rlimits*, использующей PAM для авторизации пользователей, запрашивающих приоритет реального времени. Это означает, что отпала необходимость запускать Jack или аудиопрограммы от имени суперпользователя. Примечательное свойство систем с низкой латентностью – это некоторое упрямство приоритизированных приложений и нежелание быстро обновлять экран, так как они ни с кем не хотят делиться ресурсами процессора. Зато такая система выдает монолитную аудиозапись.

Итак, моя новая аудиосистема была почти готова. Оставалась только оптимизировать жесткий диск, чтобы он справлялся с аудиопотоками реального времени. Хотя приоритет для диска и был установлен, сам диск все же оставался в стандартном режиме, что приводило к зависанию при попытке одновременной записи более четырех каналов. Несколько пассов с *hdparm* для установки 32-битного режима (**-c 1**), UDMA (**-d 1 -x 66**) и мультисекторной передачи (**-m 16**) завершили работу. Если бы я захотел включить в работу все восемь каналов, то мне, вероятно, потребовался бы RAID или, как минимум, SATA-диск, чтобы справиться с записью/чтением потока аудиоданных.



Часть 2 Процесс пошел

На что же способна свободная студия?

Чтобы протестировать студию, я пригласил многообещающую группу под названием The Mandibles для записи нескольких дорожек. Там были три гитариста, один клавишник, двойной бас, саксофон, два вокалиста, труба и ударник. Было непросто, ведь в моей студии нет отдельных боксов для записи, поэтому я нашел несколько тихих мест в своей кухне и установил в них микрофоны: Shure SM58 для струнных, SM57 для ударных и баса, и AKG для вокала и духовых.

Ardour весь вечер безупречно записывал четыре канала разом в поразительно чистый 32-битный 96-килогерцовый файл WAV-формата. Я легко перемещал записи между каналами и вставлял новые отрезки без ущерба для производительности. Трудности появились лишь при попытке одновременной записи шести каналов (очевидная проблема быстродействия жесткого диска), даже на низких битрейтах. Мне хотелось записать для каждого микрофона отдельный канал в *Ardour*, чтобы получить максимальную гибкость при дальнейшей обработке звука. Вместо этого все микрофоны были подключены к микшеру и сгруппированы на входы звуковой карты как две стереопары.

Имея в распоряжении всего четыре канала, я мог быстро переключаться между дорожками и добавлять эффекты без замедления. С быстрым жестким диском *Ardour* работает более гладко и никогда не запинается, даже при шестиканальной записи. Отладив процесс аудиозаписи, я предполагаю записывать все восемь каналов одновременно.

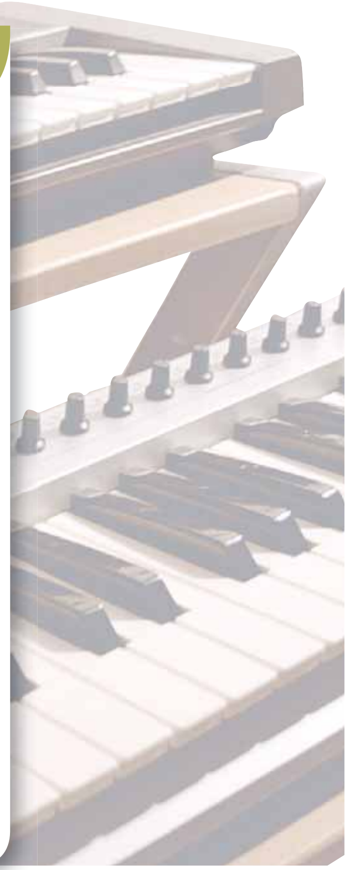
Вместо микширования в *Ardour*, я воспользовался микшером и направил все аудио-каналы обратно на звуковую карту для записи. Таким образом, я мог пользоваться отдельным эквалайзером для каждого канала микшера и аппаратным компрессором вокала. Проще делать это физическими органами управления, а возможность расположить все дорожки рядом облегчает наблюдение за настройкой.

Все программы моей новой системы прекрасно взаимодействовали, хотя пользователям коммерческого ПО интеграция между ними может показаться не столь тесной, как им привычно. Я пользовался многими разными программами для решения разных задач, что делало переключение между дорожками трудоемким и подчас бессистемным. Тем не менее я сумел открыть файл *Hydrogen* в *Rosegarden* и увидеть автоматически установленную связь между двумя программами. Мне осталось лишь добавить инструменты. В будущем интеграцию между Linux-приложениями должен углубить проект LASH – см. http://savannah.nongnu.org/projects/lash.

Гармония ПО

В моем случае, Linux в деле звукозаписи имел успех. Настройка проще, чем у Windows, меньше перезагрузок и не требуется загрузка драйверов (Windows не включает драйверы и ПО в стандартную инсталляцию). А главное, не понадобилось ничего устанавливать необычным, особенным способом. Нельзя не упомянуть о жестких ограничениях на число MIDI- и аудиопортов в Windows. Зато я потерял некоторые мои любимые проприетарные программы, и мне трудно ощущать себя столь же креативным на Linux с самого начала из-за упрощенных интерфейсов и отсутствия демо-файлов для обучения.

Ядро с поддержкой приоритетов реального времени – это здорово, хотя инструмент добавки модулей не работает, и простые добавления к ядру недоступны. При этом ничто не мешает собрать собственное ядро и добавить, например, *Xgl* для освобождения процессора от бремени отрисовки графики, или добавить привычную программу вроде *Vi* (по умолчанию в 64 Studio стоит *Nano*). Экспериментирование – одна из прелестей свободного ПО, и вы можете поискать совета и вдохновения на http://linux-sound.org, www.linuxaudio.org и http://lalists.stanford.edu. Дерзайте же – освободите внутреннее Я и возьмитесь за рукоятки. LXF



Интервью: Дэниел Джеймс



Дэниел Джеймс – директор проекта аудио-дистрибутива 64 Studio. Он учредил 64 Studio Ltd, предоставляющую услуги по разработке независимым поставщикам и поддержку пользователям в студиях (www.64studio.com). Кроме того, Джеймс владеет студией звукозаписи на основе Linux недалеко от дома, на острове Уайт.

Linux Format: **Каково основное требование к студии звукозаписи на базе Linux? Гибкость, стабильность, скорость…?**
Дэниел Джеймс: Надежность. Не просто стабильность, а полное отсутствие неполадок: необходима уверенность в том, что оснастка не подведет. При записи живой музыки второй попытки не бывает.

LXF: **Сколько дорожек вам случалось записывать одновременно?**
ДДж: На моей системе Opteron я еще никогда не приближался к физическому пределу аппаратуры. Сегодня количество каналов обычно ограничивается числом каналов аудиоинтерфейса, а не пределами программы или процессора. Теоретически, пользуясь *Ardour* и звуковой картой RME MADI, можно записывать до 64 каналов одновременно – но такое количество мало кому нужно.

LXF: **Насколько близки Linux-решения к своим проприетарным соперникам?**
ДД: Не всегда можно найти подходящий эквивалент, но для работы с музыкой средств вполне достаточно. В некоторых случаях, особенно в экспериментальных жанрах, именно проприетарному ПО недостает функциональности.

LXF: **Можно ли монтировать фонограмму с помощью Linux?**
ДДж: Легко. Имея старый компьютер со слабым процессором, можно обрабатывать дорожки по отдельности с использованием LADSPA-эффектов, например, в *Audacity*. Счастливые обладатели новейшего оборудования могут монтировать в реальном времени с помощью *Jamin*.

LXF: **Есть ли средства для компрессии, сглаживания и авторинга CD по Red Book?**
ДДж: *Jamin* обладает хорошим трехдиапазонным компрессором, а также 1,024-полосным эквалайзером с ручным подбором кривых, упреждающим ограничителем и другими инструментами. Сглаживание вывода можно делать из *Ardour*, *Audacity* или прямо из Jack. Что касается экспорта аудио в Red Book, то сегодня люди в основном отправляют WAV-файлы на CD-R дубликаторам и предоставляют им заботу об остальном, поэтому я никогда не слышал, чтобы кто-нибудь жаловался на нехватку свободного ПО для записи DDP-файлов на Exabyte-магнитофон.

LXF: **Вы считаете, свободное ПО вносит свой вклад в коллективизацию производства музыки?**
ДДж: Определенно, хотя существует риск чрезмерно увлечься созданием инструментов и забыть о самой музыке. Правда, хорошо написанная программа красива сама по себе, а аудиосообщество Linux гораздо дружелюбнее к новичкам, чем некоторые музыкальные союзы.

› Полный текст интервью см. на www.linuxformat.co.uk/mag/james.html.



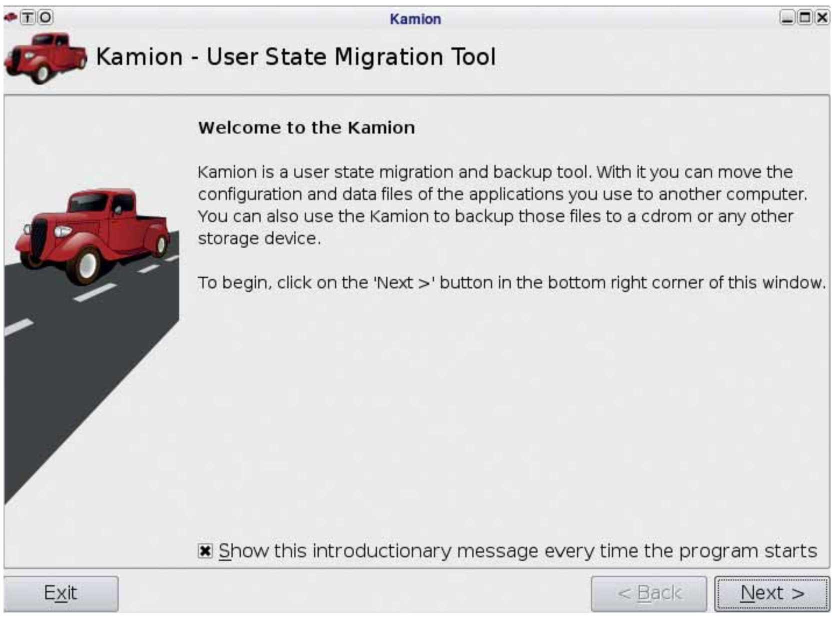
KDE 4 на колесах

Переезд на новую квартиру – настоящая головная боль. Миграция с одного компьютера на другой – тоже малоприятное занятие. **Натан Сандерс** расскажет нам о новом приложении KDE, которое облегчит трудности переезда.

› Старенький пикап может напомнить вам шоу «*The Beverly Hillbillies*», но методы перевозки рухляди в Kamion впечатляюще современны.

KDE встречается чаще, чем вы можете предположить. Некоторых может отпугнуть большое количество разнообразных опций, но фанаты-настройкокопатели вряд ли будут чувствовать себя так же комфортно в других оконных средах. Теперь, благодаря новому инструменту переноса пользователей Kamion стать таким фанатом под силу каждому…

Тем, кто использует KDE на нескольких компьютерах, наверняка приходилось вручную задавать одни и те же настройки среды на разных системах. Кроме этого, время от времени вам наверняка случается создавать резервные копии важных данных и хранить их на всех доступных ПК. И в том, и в другом случае поможет *Kamion* (одно из новых приложений, появившихся в KDE 4). Используя встроенную базу данных и взаимодействуя с пользователем, эта программа собирает всю необходимую информацию (личные файлы и настройки приложений). После этого формируется архив, который можно перенести на другой компьютер и там распаковать. Таким образом, это средство миграции пользователей позволит вам перенести все настройки среды KDE с одного компьютера на другой.



Простое копирование каталога **/home** неэффективно и потребует много времени на перенос объемного и ненужного кэша браузера. К тому же некоторые приложения могут вовсе не запуститься на другой системе из-за конфликтов версий, что изрядно потреплет вам нервы. *Kamion* (в переводе с сербского «грузовик») использует более продуманный подход. Он знает, что нужно и что не нужно брать с собой в новый дом. В идеальном случае каждое приложение должно добавить в базу данных *Kamion* информацию о том, какие файлы необходимы для его работы и с какими версиями программы они совместимы, а

> «Kamion позволит перенести все настройки среды KDE с одного компьютера на другой.»

какие файлы содержат временные данные и не имеют большого значения. *Kamion* использует эту информацию для создания архива, после восстановления которого на другом компьютере мы получим окружение, полностью повторяющее оригинал.

Впрочем *Kamion* пригодится даже тем, кто не собирается работать в KDE 4. Ведущий разработчик проекта Иван Кукич [Ivan Cukic] создавал его в рамках The Season of KDE, ответвления Google Summer of Code. Он разрабатывает *Kamion* в двух направлениях: а) простой графический интерфейс и б) сервисная библиотека *libKamion*, которая может использоваться оболочками, ориентированными на другие оконные среды или даже отдельные приложения.

Экскурсия

Это краткое руководство по *Kamion* предназначено для смельчаков, пользующихся текущей версией KDE 4, которая еще находится в разработке. Для тех, кто захочет установить KDE 4, на страничке http://developernew.kde.org/Getting_Started приведены инструкции по установке. Даже если вы просто хотите посмотреть, как выглядит и работает *Kamion*, эта статья будет для вас полезной. Обратите внимание, что для создания резервной копии с помощью *Kamion* нам потребуется найти XML-файлы состояния какого-нибудь приложения.

Пошаговое руководство на странице 45 наглядно демонстрирует все возможности текущей версии приложения, но это лишь небольшая часть того, что может предоставить библиотека *libKamion*. Одной из следующих задач разработчика является механизм добавления в архив

.*Kamion* произвольных файлов и каталогов, который позволит вам сделать резервную копию файлов из папки **Documents** или приложения, которое не распознается автоматически. Что же касается механизма постоянной синхронизации настроек системы на двух компьютерах, то его разработка пока не планируется. Кукич считает, что эта задача выходит за рамки функционала его приложения, но уверен, что разработать такой компонент поверх *libKamion* будет достаточно просто.

Кое-что для разработчиков

Будущее *Kamion* целиком и полностью зависит от связи этого приложения с другими. С приложениями, которые предоставят *Kamion* файл XML с описанием своего текущего состояния (эти файлы будут скомпилированы в базу данных *SQLite 3*, которая выступает лишь как кэш для них), никаких проблем не будет. Файлы конфигурации и данных для таких приложений будут полностью восстановлены на другом компьютере. Настройки остальных приложений придется переносить вручную или попросту отказаться от этой затеи.

К счастью, обеспечить поддержку некоторого приложения в *Kamion* очень просто. Файл XML, который должно предоставить это приложение, должен отвечать стандарту Freedesktop.org (который сейчас обсуждается) и содержать следующие элементы:

- Название приложения.
- Общая категория для этого приложения, например, 'web.browser'.
- Краткое описание приложения. Например, 'IM program'.
- Иконка приложения. Может быть иконкой этого приложения с рабочего стола.
- Файл использования (existence test). Это файл, который создается при запуске приложения. Если существование XML-файла говорит о том, что пользователь установил приложение, то существование файла использования приложения означает, что пользователь запускал его и вероятно захочет сделать копию его состояния.
- Список ресурсов приложения. Список всех файлов конфигурации и данных, которые использует приложение. Каждый элемент списка должен иметь уникальный идентификатор (например, 'kopete.history'), название (например, 'chat history'), полный путь до файла ресурса (например, **$HOME/.kde/share/apps/kopete_history**) и описание (например, 'Record of IM conversations').
- Версия приложения или конфигурационного файла. Если указана версия приложения, то в случае ее отличия, *Kamion* предупредит пользователя о несовместимости. Если указана версия файла конфигурации, *Kamion* спокойно выполнит обмен файлов между различными версиями приложения.

Последняя возможность *Kamion* может быть расширена. Сейчас *Kamion* только выдает предупреждение в случае, когда версии конфигурационного файла в резервной копии и на другом компьютере несовместимы. В будущем *Kamion* может использовать специальные скрипты для конвертирования конфигурационных файлов различных версий. Эти скрипты будут поставляться приложениями. Еще проще возложить эту задачу на сами приложения. В этом случае им следует «разрешить» *Kamion* использовать конфигурационные файлы, отвечающие различным версиям приложения.

Стандартизация

Стандарты Freedesktop.org уже предусматривают наличие у каждого приложения файлов для связи с рабочим столом. Это файлы **.desktop**, содержащие информацию о названии приложения, которое появится в меню Программы, и параметрах работы приложения (например, перечень команд для его запуска). Поэтому добавление другого стандартного файла – файла состояния приложения для *Kamion* – не слишком суровое требование, и даже не слишком оригинальное.

К сожалению, большинство участников скромной команды разработчиков *Linux Format* ничего не слышали о *Kamion*. Даже те из них, кто разрабатывал приложения для KDE 4, не подозревали о его существовании. Поэтому сейчас этому приложению еще требуется некоторое продвижение. Чтобы обеспечить полную совместимость с другими приложениями, потребуются напряженная работа и тесное взаимодействие всего сообщества. Стандартизация Freedesktop.org – первый шаг на этом пути. »

Перенос пользователей в Windows

Первое средство для миграции пользователей, мастер переноса файлов и установок – *Files And Settings Transfer Wizard* (FAST), появилось в Windows XP. Это средство обеспечивало перенос установок с Windows 95, 98, ME и 2000 в Windows XP. Для корпоративных пользователей была выпущена расширенная версия продукта – *User State Migration Tool* (USMT). В Windows Vista используются новая версия USMT 3.0 и Windows Easy Transfer как замена FAST.

В отличие от *Kamion*, в этих программах акцент смещен на перенос пользователя на новый компьютер или операционную систему, а не на создание резервных копий или перенос настроек системы с одного компьютера на другой. Как следствие, они содержат ряд интересных возможностей, в которые входит быстрая передача и восстановление данных на другом компьютере через соединение по USB, Ethernet и другим интерфейсам. Есть возможность передать профили пользователей и другие низкоуровневые подробности, что может представлять определенную сложность в Unix-системах.

Как и в *Kamion*, в этих программах производится резервное копирование и восстановление конфигурационных файлов и личных данных пользователя. Файлы от некоторых программ распознаются автоматически (по большей части, это продукты Microsoft). *Kamion* разрабатывается не так давно и не обеспечивает поддержки многих приложений, но использование XML-файлов состояний, стандартизированных FreeDesktop.org, поможет обеспечить прекрасную совместимость со всеми приложениями.

Резервирование или миграция?

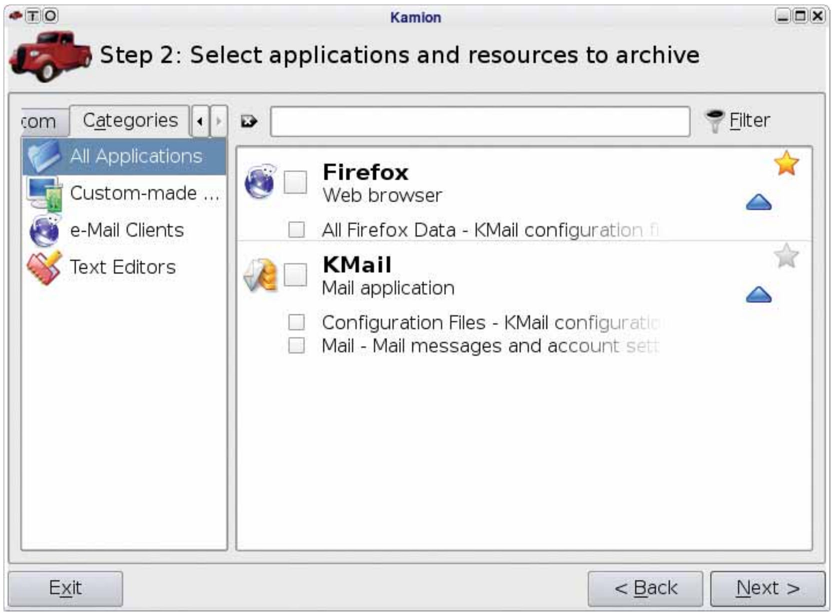
Мы все используем средства создания резервных копий, но в чем их отличие от средств миграции пользователей?

Задача	Средство резервирования	Средство миграции пользователей
Архивирование личных файлов пользователей	✓	✓
Сжатие архива	✓	✓
Архивирование конфигурационных файлов приложений	✓	✓
Предупреждение о несовместимости версий конфигурационных файлов	☒	✓
Игнорирование временных файлов и кэша браузера	☒	✓
Предотвращение восстановление файлов неиспользуемых программ	☒	✓



» Хотя Иван Кукич разрабатывает *Kamion* для KDE 4 и операционной системы Linux, это приложение потенциально может использоваться и на других Unix-системах, таких как *Solaris* и BSD. Единственная платформа, поддержку которой Кукич не планирует – это Windows. Версия программы под Windows может потребовать значительных изменений в коде, и Кукич производить их не хочет. Он советует разработчикам «использовать модель хранения данных Unix (файлы и каталоги в домашнем каталоге пользователя, название которых начинается с точки)».

После объявления Trolltech о том, что библиотека *Qt 4*, которая лежит в основе KDE 4, будет доступна пользователям Windows по лицензии GPL, разработчики KDE успешно переносят многие приложения KDE на эту платформу. Распространенность операционной системы от Microsoft представляет приложениям KDE вполне понятные перспективы, а также сможет привлечь новых разработчиков. Однако некоторые приложения - под Windows и поэтому не будут переноситься. Это оконный менеджер *KWin*, и, возможно, *Kamion*. В самом деле, аналоги *Kamion* в Windows уже существуют (*см. врезку «Перенос пользователей в Windows»*).



› **Интерфейс *Kamion* в KDE 4 довольно симпатичен. Это анимация при раскрытии списков и плавное исчезновение строк текста, которые не умещаются в окне.**

За пределами KDE 4

Хотя Кукич и не собирается создавать версию *Kamion* под Windows, другим разработчикам не составит труда сделать это за него. Интерфейс *Kamion* на *Qt 4* – это лишь графическая оболочка для средства миграции пользователей, вся реализация которого заключена в сервисной библиотеке *libKamion*. Кукич с самого начала решил сохранять разделение между серверной библиотекой и графическим интерфейсом, чтобы другие разработчики смогли создавать свои собственные оболочки.

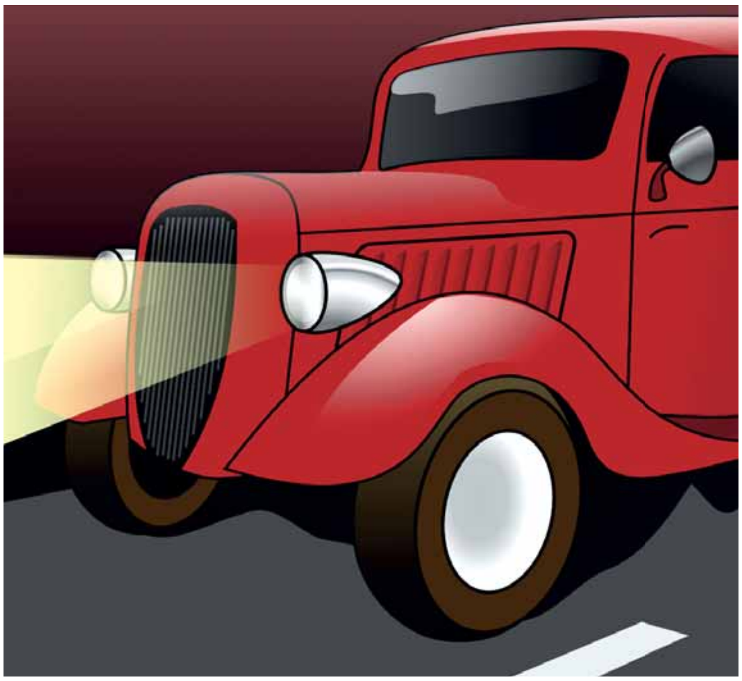
В качестве клиентов библиотеки *libKamion* могут выступать даже отдельные приложения. Например, пользователи почтового клиента, использующего эту библиотеку, смогут создавать резервные копии своего почтового ящика. Кукич уже написал простейшую консольную оболочку, использующую *libKamion*.

Разработчики могут воспользоваться всеми преимуществами библиотеки уже сейчас – Кукич завершил ее разработку в августе прошлого года. Сервисная библиотека предоставляет разработчикам весь необходимый функционал для создания средства миграции пользователей. Она содержит инструменты для взаимодействия с базой данных, сбора и архивирования необходимых файлов и восстановления файлов из архива. Сейчас библиотека использует базу данных *SQLite 3* и *Qt 4*. В дальнейшем, чтобы устранить зависимость от *Qt 4*, Кукич надеется выпустить версию программы, которая будет использовать Xerces-C для синтаксического анализа XML и PStreams для управления процессами.

Разработчики приложений для Gnome, вероятно, создадут собственное средство миграции пользователей. Впрочем, одни и те же технологии часто используются в различных оконных средах (например, библиотека для просмотра PDF Poppler и система обмена сообщениями D-BUS). Поскольку весь код приложения выпущен по лицензии GPL 2, разработчики могут использовать его практически без ограничений.

Успех *Kamion* невозможен, если каждое приложение не предоставит ему информацию о себе в виде XML-файла. И только от числа разработчиков, готовых принять участие в интеграции своих проектов с *Kamion*, зависит, станет ли это приложение неотъемлемой частью KDE 4. Тем не менее, если разработчики захотят помочь друг другу, *Kamion* может стать «островом Эллис» для KDE. [Остров Эллис – небольшой остров близ г. Нью-Йорка. В 1892–1943 – главный центр по приему иммигрантов в США. С 1965 стал частью Национального памятника «Статуя свободы» – прим. перев.]

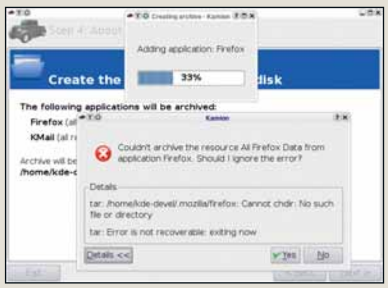
Чтобы принять участие в проекте, зайдите на страничку http://Kamion2.sourceforge.net. LXF



Файл .kamion

Резервная копия хранится в единственном файле **.kamion**, который представляет собой не что иное, как обычный архив tar, содержимое которого (файлы данных и конфигурации) имеет определенную структуру. Внутри этого архива можно найти файл **Manifest.xml** с полным списком ресурсов архива и указанием на то, каким приложениям они принадлежат. Сами файлы ресурсов упакованы и хранятся в файлах типа **appname.resource name.tar.bz2**. В архиве также содержится файл **Wishlist.xml**, который содержит список всех ресурсов, выбранных пользователем. Различия между этими файлами могут быть только в том случае, когда при попытке включения файла в архив произошла ошибка.

Файл **Wishlist.xml** пока не используется приложением, но может оказать вам неоценимую услугу, когда вы начнете разбираться, куда делись все те почтовые сообщения, которые вы год как скопировали на резервный носитель.



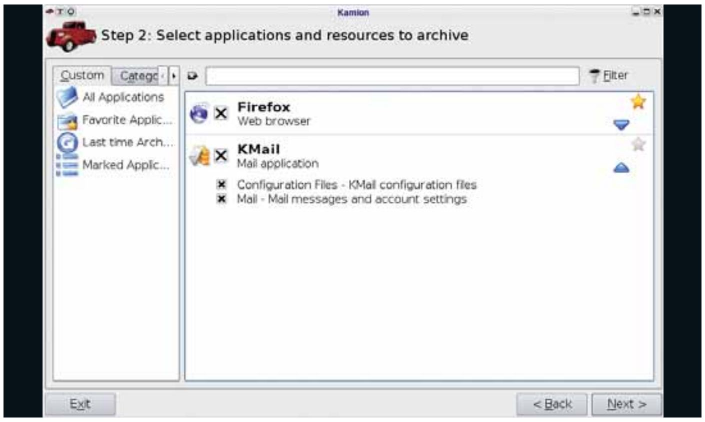
› **Список ресурсов в файлах wishlist.xml и manifest.xml может различаться. Например, если удалить каталог с данными *Firefox* до создания архива, *Kamion* сообщит нам, что произошло, и предложит продолжить создание копии без данных *Firefox*.**

Шаг за шагом: Переносим настройки KDE с помощью Kamion



1 Создаем резервную копию

Выберите Pack The Data, чтобы создать резервную копию, или Restore The Data, чтобы восстановить данные. Диалог Settings позволяет указать местоположение файлов настройки *Kamion* и программу для записи компакт-дисков.



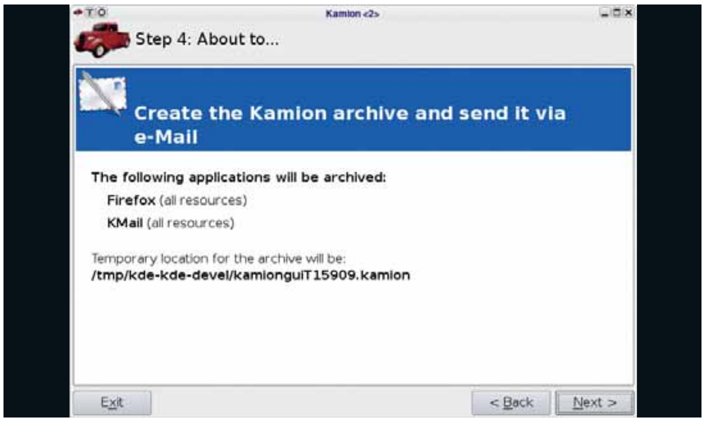
2 Выбираем приложения и ресурсы

В окне отобразится список приложений, которые поддерживаются *Kamion*. С помощью стрелок в правой панели можно просмотреть полный список ресурсов для каждого приложения и выбрать те, которые будут включены в архив. Вкладки слева и строка фильтра справа помогут задать условия поиска приложений.



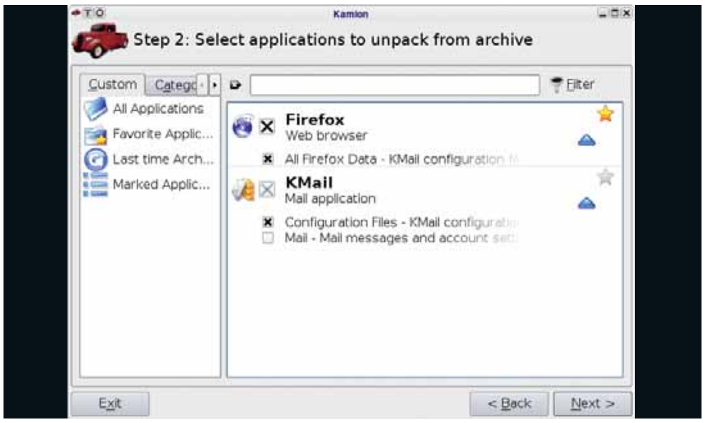
3 Сохраняем резервную копию

Кнопка Save To Disk вызовет стандартный диалог, с помощью которого можно сохранить архив на диске. Кнопка Send By Mail позволит вам отослать архив по почте посредством указанного ранее клиента. Архив можно также записать на CD.



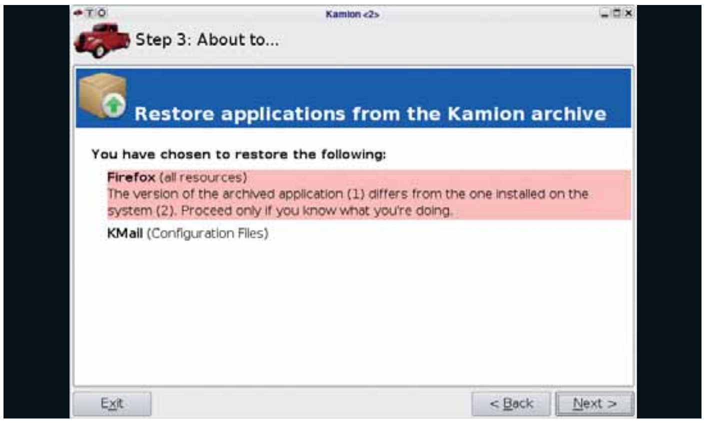
4 Миссия выполнена

Kamion отобразит список ресурсов создаваемого архива, и у вас будет последний шанс изменить его, нажав кнопку Back. Запомните: как и во всех приложениях KDE, вы можете использовать виртуальную файловую систему KIO, чтобы сохранить файл на удаленном узле.



5 Восстанавливаем данные

Диалог восстановления данных из архива мало чем отличается от диалога его создания. *Kamion* отобразит полный список ресурсов архива, и вы сможете выбрать, какие из них нужно восстановить, а какие нет. Вы можете воспользоваться фильтром приложений по ключевым словам, категориям, избранному, истории архивирования и т.п.



6 Неизбежное

Конечно, все может пройти и не совсем гладко. Данное сообщение об ошибке вызвано более новой версией *Firefox*. Если мы до конца не уверены в том, что каталоги данных разных версий *Firefox* полностью совместимы, лучше вернуться на предыдущий шаг и отменить восстановление. Если бы *Firefox* не был установлен совсем, *Kamion* тоже выдал бы соответствующее предупреждение.



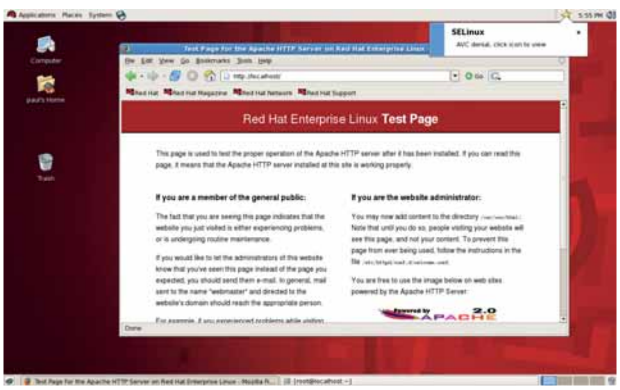
Neil Bartlett

Доступно о SELinux

Вы запросто достигнете безопасности правительственного уровня, если у вас хватит терпения постигнуть ее суть. **Пол Хадсон** упростит эту задачу...

Red Hat Enterprise Linux 5 выпущен, и, как и следовало ожидать, многие в восторге от интегрированной виртуализации *Xen*. Но в недрах системы скрыта другая, столь же продвинутая технология, о которой вы наверняка уже слышали: Security Enhanced Linux, или, сокращенно, SELinux. Подобно драйверу звуковой карты или *OpenOffice.org*, SELinux – один из тех компонентов Open Source, о которых никто даже не задумывается: он работает в фоновом режиме, и вы вспоминаете о нем, только когда возникают проблемы.

В ранних дистрибутивах, содержащих SELinux, например, Fedora Core 6, SELinux был скорее помехой, чем подмогой, и его норовили отключить, поскольку он осложнял повседневную работу настольного ПК. Но RHEL 5 включает свежую версию SELinux, которая, сохранив исходный принцип SELinux, обзавелась множеством инструментов, упрощающих понимание ее установки и использования. В нашей статье мы познакомим вас с возможностями SELinux и покажем, как освоиться с ним в RHEL 5. Мы использовали свежую инсталляцию RHEL 5, прихватив дополнительный пакет Web Server, чтобы использовать *Apache* для теста на выживаемость. CentOS 5 – фактически RHEL 5, только без товарного знака – тоже включает поддержку SELinux, а еще вы сможете следовать большей части нашего урока на Fedora Core 7. Технологии, описанные здесь, применимы и для других сетевых демонов.



› *Apache* может просто молча отказаться читать index.html, тогда как SELinux предупредит вас о произошедшей неприятности.

Безопасный по умолчанию

SELinux был изобретен в Национальном Агентстве безопасности США для удовлетворения потребности в... впрочем, вам, наверное, неинтересна история SELinux и всякая ерунда. Лучше мы вам покажем, как SELinux защищает вас 24 часа в сутки, 7 дней в неделю, независимо от того, осознаете вы это или нет.

Откройте терминал, введите

SELinux до Unix

SELinux не может разрешить то, что запрещают стандартные права доступа Unix. Например, если файл **index.html**, с которым мы работали в этой статье, имеет правильный контекст SELinux для чтения демоном Apache, но не имеет прав доступа Unix на чтение для пользователя, от имени которого работает *Apache*, ничего не получится. SELinux служит для дополнения прав доступа Unix, а не для их отмены!

```
su -
vim index.html
```

затем для перехода в режим вставки нажмите i и введите:

```
Hello, world!
```

Для сохранения файла нажмите Escape, затем введите **:wq**. Вы должны снова очутиться в командной строке [вы также можете использовать любой другой привычный вам редактор, – прим. ред.]. Теперь выполните следующие две команды:

```
Apachectl start
mv index.html /var/www/html
```

Это вполне обычная штука: мы создали HTML-файл с некоторым содержимым, запустили *Apache*, затем скопировали новый файл в каталог, который используется *Apache* для обслуживания web-страниц. Но сейчас вы увидите, насколько педантично печется SELinux о вашей безопасности! Попробуйте запустить *Firefox* и указать ему адрес http://localhost, чтобы он запросил данные с локального сервера *Apache*. Вместо того, чтобы увидеть 'Hello, world!' вашего **index.html**, вы увидите приветственную страницу Red Hat, установленную по умолчанию. Пока вы оправляетесь от секундного замешательства, в правом верхнем углу экрана появится пузырик с сообщением: 'SELinux: AVC denial, click icon to view.' [Запрет AVC, щелкните на иконке для просмотра.]

Если 'AVC' для вас пустой звук (оно расшифровывается как Access Vector Cache, хотя от этого вам вряд ли полегчало), то слово 'denial' уже намекает на происшедшее: когда вы пытались загрузить страницу с *Apache*, *Apache* проверил, можно ли ему читать этот файл, и получил отказ от SELinux. Однако вернувшись к вашему окну терминала и выполнив **ls -l** в **/var/www/html**, вы увидите, что файл имеет флаги разрешения -rw-r--r--, то есть доступен всем пользователям. И что это даст? Щелкнув по значку, запустим *Setroubleshooter*, новую утилиту, предназначенную для перевода практически невразумительных сообщениях об ошибках SELinux на язык, понятный простым смертным.

В нашем случае вы увидите вверху сообщение, а внизу – его объяснение. Сводка должна звучать примерно так: 'SELinux пресек обращение пользователя /**usr/sbin/httpd** к файлу с потенциально неверным маркером (**/var/www/html/index.html**)'. Вам, может быть, это ничего не говорит, однако прокрутив информационную панель донизу, вы найдете нечто более осмысленное по сравнению с «сырым» сообщением аудита:

```
avc: denied { getattr } for comm="httpd" dev=dm-0 egid=48 euid=48
exe="/usr/sbin/httpd" exit=- 13 fsgid=48 fsuid=48 gid=48 items=0
name="index.html" path="/var/www/html/index.html"    pid=2877
scontext=user_u:system_r:httpd_t:s0 sgid=48 subj=user_u:system_r:
httpd_t:s0 suid=48  tclass=file tcontext=user_u:object_r:user_home_t:s0
tty=(none) uid=48
```

И сырое сообщение, и аккуратно разобранная версия *Setroubleshooter* советуют нам обратить внимание на контекст, так как для SELinux все – и процессы, и файлы – имеет 'контекст' безопасности, используемый как маркер. К счастью, *Setroubleshooter* сумел опознать данную конкретную проблему и говорит, что использование *mv* для копирования файлов (что мы пытались сделать после **Hello World!**) часто вызывает проблемы, поскольку не меняет контекст безопасности источника.

Иными словами, проблема такова: SELinux полагает, что файлы из нашего домашнего каталога не должны обслуживаться в качестве web-страниц – и правильно полагает: если вдруг ваш сервер *Apache* будет взломан, вы не захотите, чтобы ваши личные файлы были доступны всему миру. Вот это и обеспечивает SELinux, последняя линия обороны вашего сервера: у вас все-таки есть возможность выжить, даже если в вашу систему засадили удаленный эксплойт.

Опция контекста -Z

Если SELinux основывается на контекстах безопасности файлов и процессов, как мы можем узнать, что это за контексты? Более того, как можно их поменять? Оказывается, существует универсальная опция, дополняющая приложения командной строки, и она выдает контекстную информацию. Назовем ее опцией контекста, поскольку все приложения, используемые вами – *ps*, *ls*, и т.п. – имеют эту специальную опцию **-Z** для вывода контекстной информации. Обратите внимание, что здесь используется именно заглавная **Z** – к ней сложно подобрать смысл, и, видимо, именно поэтому она годится для всех приложений!



› **SELinux Troubleshooter придуман для распознавания типовых проблем и помощи в их исправлении. Его советы не всегда идеальны, слепо следовать им не стоит!**

Используя **-Z**, можно извлекать контекстную информацию различных объектов. Например, перейдите в каталог **/var/www** и выполните **ls -Z**. Вы увидите следующее:

```
[paul@localhost www]$ ls -Z
drwxr-xr-x root root system_u:object_r:httpd_sys_script_exec_t cgi-bin
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t error
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t html
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t icons
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t manual
drwxr-xr-x webalizer root system_u:object_r:httpd_sys_content_t usage
```

А если вы сделаете то же в каталоге **/var/www/html**, вы увидите вот это:

```
-rw-r--r-- root root user_u:object_r:user_home_t index.html
```

Контекст **user_home_t** сообщает, что **index.html** принадлежит домашнему каталогу пользователя, а это значит, что *Apache* не будет его обслуживать. Взамен ему следует иметь контекст **httpd_sys_content_t**, то есть контекст по умолчанию для **/var/www/html**. В этом – основное отличие *mv* от *cp*: *mv* перемещает файлы, и они сохраняют свой прежний контекст; cp создает новый файл, и он наследует права доступа каталога назначения.

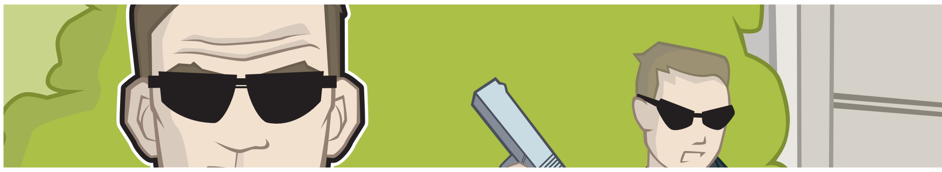
Горю можно помочь, присвоив файлу **index.html** контекст **httpd_sys_content_t** командой *chcon*, примерно так: »

SELinux против AppArmor

SELinux часто критикуют за то, что, в отличие от *AppArmor* (система безопасности Novell, приписывающая профили различным программам), здесь не используется режим обучения для выяснения настроек, требуемых для правильной работы системы. Например, для настройки в *AppArmor* вашего сервера *Apache* вам просто следует поместить его в режим наблюдения, затем загрузить все страницы (и, в частности, все исполняемые скрипты), и он вам покажет, в какой степени сервер требует доступа к машине.

Однако способ, применяемый *AppArmor*, имеет недостаток: *AppArmor* полагает, будто вам лучше знать, что безопасно для вашего сервера, а что нет. Вам не составит труда обучить его дурному поведению, и тогда он уже не станет извещать вас о проблемах. Например, если ваш сервер настроен небезопасно, но вы заявили *AppArmor*, что это – штатное поведение, не видать вам потом предупреждений!

В случае с SELinux, Red Hat обеспечивает предустановленный набор правил для всех общепринятых сервисов, гарантирующих, что вы в безопасности уже по умолчанию. Имеется специальная утилита под названием *audit2allow*, позволяющая создать правила на основе AVC-запретов.



```
chcon -t httpd_sys_content_t /var/www/html/index.html
```

Параметр **-t** указывает тип **httpd_sys_content_t**, так что, введя **ls -l**, вы увидите следующее:

```
-rw-r--r-- root root user_u:object_r:httpd_sys_content_t index.html
```

Если *Firefox* у вас все еще запущен, убедитесь, что сейчас доступ к **index.html** уже имеется, поскольку SELinux не против, чтобы *Apache* читал принадлежащее ему содержимое. Если нужно изменить несколько файлов, *chcon* используется так же, как *chmod*: укажите несколько имен файлов и примените регулярное выражение наподобие **index.*** или параметр **-R** обработки вложенных каталогов.

Программа *restorecon* пользуется тем, что SELinux считает файлы в **/var/www** имеющими контекст **httpd_sys_content_t**, и может скопом поменять их контекст на принятый для данного каталога по умолчанию. Например, следующая команда присвоит всем файлам в каталоге **/var/www/html** контекст **httpd_sys_content_t**, заходя внутрь подкаталогов (опция **-R**) и выводя имена файлов, подвергшихся изменениям **(-v)**:

```
restorecon -v -R /var/www/
```

Проверить контекст по умолчанию для других файлов можно командой

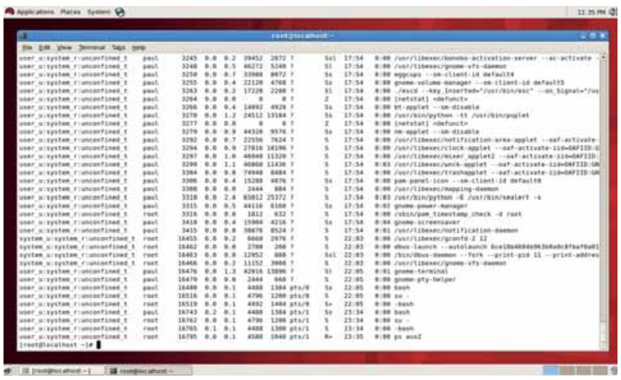
```
semanage fcontext -l
```

SELinux использует регулярные выражения как шаблон имен файлов. Например, правило, с которым мы работали до сих пор, выглядит как

```
/var/www(/.*)? all files system_u:object_r:httpd_sys_content_t:s0
```

Опцию **-Z** можно использовать где-нибудь еще, например, с командой ps. Выполнив ее, вы заметите, что большая часть запущенных в вашей системе приложений выполняется с контекстом **unconfined_t**. Исключения – *GDM*, *X*, *Sendmail*, *cupsd* и различные сетевые службы, плюс некоторые ключевые службы, важные для функционирования системы. Это потому, что большинство запускаемых вами приложений используют ваш уровень привилегий, то есть смогут влиять только на файлы с тем же Unix-уровнем доступа, равно как и с тем же контекстом SELinux.

> «Будьте бдительны, используя SMT: любые сделанные изменения тут же вступают в силу.»



› **Многие программы командной строки обзавелись ключом -Z для вывода информации SELinux. Перед вами результат выполнения команды ps с параметрами 'auxZ'.**

Контроль доступа к порту

Мы увидели, как SELinux защищает ваши файлы, но он также защищает вашу сеть путем ограничения числа портов, к которым может обращаться сетевой демон. Полный список ограничений на порты можно получить по команде

```
semanage port -l
```

Все, что не попало в выведенный список, можно классифицировать следующим способом: если это 1023 или меньше, порт классифицируется как **reserved_port_t**; иначе он относится к **port_t**. Вы должны увидеть в списке также и **http_port_t**; если нет, попробуйте отфильтровать список с помощью **semanage port -l | grep http**. SELinux дозволяет *Apache* выполняться только на HTTP-портах, то есть разрешены порты 80, 443 и прочие. Если *Apache* попытается обслужить страницы через другой порт – даже разрешенный в обычном режиме, будучи больше, чем 1023 – SELinux отловит это и пресечет.

Давайте проверим это примером из реальной жизни. Пусть кому-то удалось перенастроить *Apache*, и он стартует с порта 3306, а не с 80. А 3306 – это порт *MySQL*, то есть вполне законный порт для соединения с сервером. Мы можем имитировать такую ситуацию, подредактировав файл **/etc/httpd/conf/httpd.conf** и изменив следующую строку:

```
Listen 80
```

Вместо **80** поставьте там **3306**. Сохраните файл и перезапустите *Apache* командой **apachectl restart**. Как и ранее, будет задержка на пару секунд перед тем, как всплывет пузырь запрета AVC: SELinux недоволен тем, что *Apache* хочет занять порт *MySQL*, и совершенно правильно говорит, что это, быть может, сигнал о попытке вторжения.

Помните, что для прочной безопасности сети главное – держать глубокую оборону: можно иметь брандмауэр по периметру сети, плюс другой, запущенный на ваших индивидуальных серверах; и, конечно же, права доступа системы Unix остановят множество проблем. Однако наличие SELinux в этой смеси добавит еще один уровень надежности и сможет пресечь на корню опасный взлом, упущенный другими. Не верите нам – послушайте Марка Кокса [Mark Cox] из команды безопасности Red Hat: «Мы включили SELinux в установку по умолчанию не без причины; не отключайте ее! В 2005 году червь Linux/Lupper использо-

Говорит Дэн Уолш

Совет от главы проекта SELinux.

«Подгружаемые модули политики позволяют администраторам легко изменять политику SELinux для их системы. SELinux иногда отказывает в доступе к ограниченному домену, а администратор считает, что зря, или, как часто бывает, просто хочет поработать. Например, политика не позволяет выполняться демону *Avahi*, поскольку ему требуются способности, о которых автор политики не знал. Для постройки и установки модуля политики администратор может использовать утилиту *audit2allow*. Чтобы изменить политику SELinux на своих машинах, можно выполнить следующие команды:

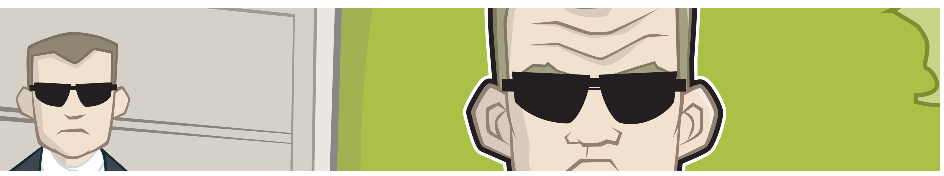
```
# grep avahi /var/log/audit/audit.log |
audit2allow -M myavahi
# semodule -i myavahi.pp
```

Политика будет изменена и останется в силе даже после перезагрузки, а также при последующих обновлениях. Разумеется, администратору следует сообщить о недочете автору политики.»



› **В своем блоге на LiveJournal Дэн публикует интересные советы и обновления SELinux.**





вал дыры в некоторых PHP-приложениях. Политики SELinux, установленные по умолчанию на Red Hat и Fedora, блокировали этого червя.»

Взгляд с высоты

Сейчас, когда вы имеете представление о методах фоновой работы SELinux, мы познакомим вас с командой *system-config-selinux*, известной также как *SELinux Management Tool* (*SMT*). Вы сможете найти его в пункте меню System > Administration > SELinux Management. Хотя она и скрывает сложности SELinux за удобным интерфейсом, работать с ней следует осторожно, так как любые сделанные вами изменения тут же вступают в силу. Существует два важнейших подводных камня:

1 Отсутствует опция отката (undo); большинство действий применяется немедленно, так что не щелкайте где ни попадя, чтобы просто «посмотреть, что будет».

2 SELinux требует времени для обработки ваших изменений, и обычно между щелчком по флажку и появлением галочки проходит несколько секунд. Если вы поторопитесь щелкнуть снова, вы попросту отмените первый щелчок, и в итоге вся операция займет еще больше времени.

Два главных раздела *SMT* – Boolean и File Labelling. Первый имеет набор опций, которые либо активны, либо нет, согласно установке флажка. Опции довольно обширны, и нередко способны полностью отключить SELinux для конкретной службы. Если вы затрудняетесь в устранении возникшей неисправности, то это и вправду может оказаться единственным подходящим решением.

Раздел File Labelling – графическая версия команды *semanage fcontext -l*, которую мы уже выполняли. Она отображает все автоматически определенные контексты для файлов и позволяет создавать новые. Но поскольку здесь доступны регулярные выражения, File Labelling полезнее банального «справочника»: вы можете видеть, как именно *restorecon* будет обрабатывать данный каталог. Опять-таки, будьте внимательны при работе с этими настройками: некоторые из них, например, перемаркировка файловой системы, занимают очень много времени, и даже могут повредить вашу систему.

И все это о контекстах

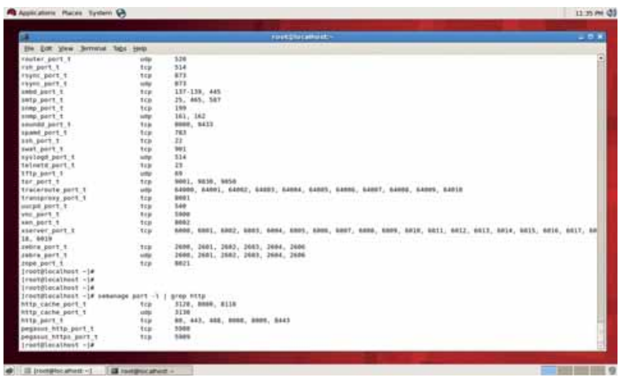
Хотя мы лишь бегло коснулись SELinux на этих нескольких страницах, надеемся, что ключевые моменты донесены до читателя:

1 По умолчанию SELinux включен, и пусть будет включен, если только из-за него не возникло серьезных проблем.

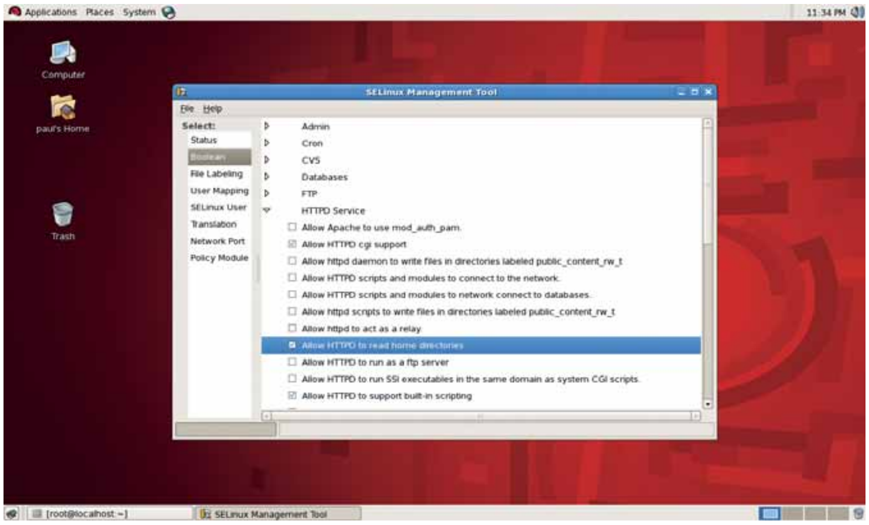
2 SELinux много сильнее, чем простая защита файлов от чтения. Процесс, защищенный SELinux, повязан по рукам и ногам, и даже будучи взломанным, не в силах причинить вреда.

3 Безопасность SELinux расположена над стандартной безопасностью Unix, так что, столкнувшись с проблемами, взгляните на контекст безопасности файла, пользователя и процесса.

4 *SELinux Troubleshooter* – самый простой способ понять сообщения AVC-запрета SELinux, и часто оттуда можно извлечь совет, как исправить проблему.



› Команда *semanage* выводит информацию об установках SELinux для контекстов порта и файла, однако вывод довольно обильный – рекомендуем профильтровать его через *grep*.



› SELinux Management Tool дает доступ к SELinux за один щелчок. Будьте с ним осторожны!

Но, возможно, самое главное вот что: SELinux – дополнительный инструмент безопасности вашего компьютера, и он работает совместно с брандмауэрами и стандартной безопасностью Unix. Он не сделает вашу систему неуязвимой, но действительно является свободным и достойным подспорьем существующих средств безопасности. LXF



Вести с полей

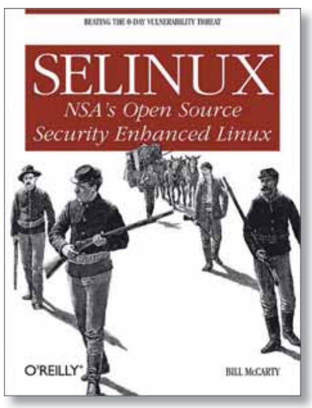
Мы попросили двух экспертов SELinux дать лучшие советы для новых пользователей SELinux.

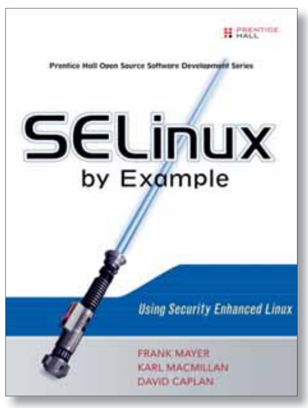
«Некоторые пользователи SELinux стараются узнать о нем как можно меньше. Подход ‘Подавлять-предупреждающие-сообщения’ может привести к ненадежности системы, ошибочно расцениваемой как надежная. А это – наихудшее состояние системы, куда хуже, чем просто ненадежная система, которая известна как надежная. Не следуйте по этому пути.»

Билл Маккарти [Bill McCarty], автор *SELinux* (O’Reilly, 2004)

«Прежде чем дать права доступа домену, следует немного подумать. Простая прогонка сообщений об отказе через audit2allow часто приводит к чрезмерно попустительской политике. Добавляйте только те разрешения, которые считаете действительно необходимыми, и обязательно затем проверяйте, работает ли приложение правильно. Используйте dontaudit для прав доступа, которые не нужны домену.»

Дэвид Капланко [David Caplanco], соавтор *SELinux By Example* (Prentice Hall, 2007)





Что за штука... D?

Вам наскучил B? Надоел C? **Пол Хадсон** проследует по алфавиту и покажет новый способ программировать.

Это была шутка насчет «B», так ведь?
Э-э-э...?

Я слышал про C. Так называется язык программирования, на котором написано ядро Linux.
Ну да. А еще – большая часть Gnome, *X.org*, *Apache* и тысячи других проектов.

Достаточно, я понял. Но неужели вы хотите сказать, что язык программирования B тоже существует?
Уверяю вас, так оно и есть. Язык B существовал еще до C, и вообще говоря, C является прямым потомком B. Такие хорошо известные элементы C, как операторы ++ и --, а также комментарии вида /* */ впервые появились в B. Но у B были свои ограничения, и поэтому его создатели, Кен Томпсон (Ken Thompson) и Дэннис Ричи (Dennis Ritchie), разработали кое-что получше.

> «Как бы мне попробовать его на практике?» «Никак. D существует только в теории.» «Что?» «Шучу, шучу.»

И назвали C?
Точно! Так как эта буква следует за B в латинском алфавите.

Выходит, шутка про C++ не такая уж и оригинальная!
Ну да, хотя любой хороший программист подтвердит вам, что правильнее было бы назвать его ++C.

Простите, не понял...
Неважно. Сегодня мы говорим про D. По этому названию вы можете видеть, что авторы рассчитывают сделать очередное улучшение над C и C++.

Примерно как Java и C#?
Ну, их создатели тоже считали, что занимаются улучшением C и C++. Но, в отличие от Java и C#, D разрабатывается как расширение C++, каким в свое время C++ был для C. Благодаря этому программистам на C++ будет легко выучить его. При этом у D есть много новых возможностей, таких как встроенная поддержка Unicode и сборка мусора.

Хм, но про Java и C# говорят в точности то же самое! Разве нам не достаточно существующих перепевок C++?
Но вы не слышали еще об одном важном отличии! Java и C# работают в виртуальных машинах и благодаря этому могут обеспечить самую причудливую защиту памяти. А D компилируется в родной машинный код, точно так же как C. Это означает, что (теоретически) код, написанный на D и на C будет работать с одинаковой скоростью. Кроме того, системные вызовы D совместимы со стандартом C, чего не скажешь о Java или C#, которым требуется специальный промежуточный слой, чтобы выбраться из виртуальной песочницы.

А какие еще функции D унаследованы от C#?
Как и C# или Java, язык D содержит string в числе базовых типов данных, а это значит, что про символьные массивы можно счастливо забыть. В стандартной библиотеке C++ уже есть концепция строкового типа, но там он прикручен сбоку и не является неотъемлемой частью языка. Например, вы пробовали сделать блок switch/case на основе строковой переменной? Нет? Это потому, что в C++ это было бы невозможно.

Превращение строк в более сложные объекты нарушает совместимость с C, но это не проблема – вы всегда можете использовать для любой D-строки метод c_str(), который возвращает обычный символьный массив, пригодный для передачи в любую функцию C.

Кстати, в D стали умнее не только символьные массивы. Фактически, все массивы в D автоматически обзавелись метаданными, такими как размер. В C эту информацию приходилось сопровождать и обрабатывать вручную.

Некоторые новые ключевые слова, такие как finally (выполнить некий блок кода в любой ситуации) или foreach (быстрый перебор элементов массива), тоже были привнесены в D из C#.

Итак, D – это некоторое расширение C++?
У D есть много функций, привнесенных из других языков программирования, но помимо этого, его разработчики воспользовались случаем избавиться от некоторых элементов C++, которые им нравились меньше всего. Например, под нож попало множественное наследование. Препроцессор тоже был уничтожен, а это значит, что макросы погибли вместе с ним. Пропали заголовочные файлы, пространства имен и ранние объявления (forward declarations). Последние оставались еще с тех времен, когда компиляторы были гораздо менее умными. Ну и, конечно, появилась автоматическая сборка мусора...

Погодите, разве вы не говорили только что, что D совместим с C?
Э нет, я говорил что D совместим с системными вызовами C, то есть вы можете пользоваться C-функциями вроде printf() в D-программах. Но, из-за большого объема добавлений и удалений, D и C не совместимы на уровне исходного кода. Это значит, что вы не можете взять C-код и скомпилировать его с помощью компилятора D. Но некоторые новшества D, вроде использования int[3] foo вместо int foo[3], в целях обратной совместимости можно и проигнорировать, так что переносите свой код на D, когда будете к этому готовы.

Если D включает в себя сборку мусора, как он может создавать столь же быстрый код, что и C?
Ну, на самом деле современные системы сборки мусора позволяют программам работать даже быстрее, чем программы с ручным управлением памятью. Да, я знаю, что это противоречит вашим интуитивным ощущениям, но подождите, и я все объясню.

Во-первых, сборка мусора запускается только тогда, когда памяти становится слишком мало, а это значит, что память не освобождается, пока в этом не появится необходимость. Для большинства программ может оказаться, что за время их работы сборка мусора не запустится ни разу – а это большой выигрыш в производительности по сравнению с C, в котором память освобождается всегда, вне зависимости от того, нужна она сейчас для чего-нибудь еще или нет.

Во-вторых, объекты C++ освобождают память с помощью вызова метода деструктора, что может сильно замедлить работу, особенно если нужно освободить несколько тысяч объектов. В-третьих, сборка мусора автоматически перегруппирует кучу, что улучшает производительность памяти.

Достаточно! Но ведь в сборке мусора должны быть и отрицательные моменты.
Да, и вполне обычные – она увеличивает объем кода, а также вполне может запуститься в тот момент, когда вам меньше всего этого хотелось бы, что может время от времени приводить к появлению маленьких

пауз в работе вашего приложения. Но эти проблемы совершенно ничтожны по сравнению с тем, сколько времени сборка мусора экономит для программиста! Последней сложностью является необходимость модифицировать ее код так, чтобы он мог запускаться на различных операционных системах.

О, вы меня просто заинтриговали этим D! Как бы мне попробовать его на практике?

Никак. D существует только в теории.

Что? Вы хотите сказать, что я провел последние 10 минут, забивая свою голову этой…

Шучу, шучу. На самом деле уже существует два работающих компилятора D – *DMD* и *GDC*. Первый из них – это официальный пример реализации компилятора, а второй – модуль к GCC. Официальный компилятор хорош своей 100% совместимостью со стандартом, чего пока нельзя сказать по *GDC*. Но при этом *DMD* имеет частично закрытые исходные тексты – часть, преобразующая D в промежуточный код, открыта, а вот компилятор промежуточного кода в тот или иной машинный – проприетарный. Сейчас эту проблему решают так: CDG использует открытый модуль, а результат его работы передает GCC, имеющему свой собственный слой генерации машинного кода. Благодаря этому *GDC* работает на большом числе платформ, включая Linux, Mac OS X, FreeBSD, Windows и AIX, как на 32-х, так и на 64-битных компьютерах.

Мне казалось, что GCC работает на большем количестве платформ.

Так и есть, но как я уже упоминал, код сборщика мусора приходится модифицировать для каждой новой операционной системы. Это не такая уж большая работа, и ее нужно сделать только один раз при портировании GCD на новую платформу, но пока ее сделали только для тех систем, которые я перечислил выше.

Вам пора привести список ссылок, поскольку доступное место уже кончается.

Хе, вовремя подмечено! О'кей, если вы хотите побольше узнать про D, зайдите на его официальную домашнюю страницу по адресу **www.digitalmars.com/d/index.html**. Вы можете скачать компилятор *DMD* на этой же странице (посмотрите под меню Tool в левой ее части). Компилятор *GDC* расположен на SourceForge: **http://dgcc.sourceforge.net**.

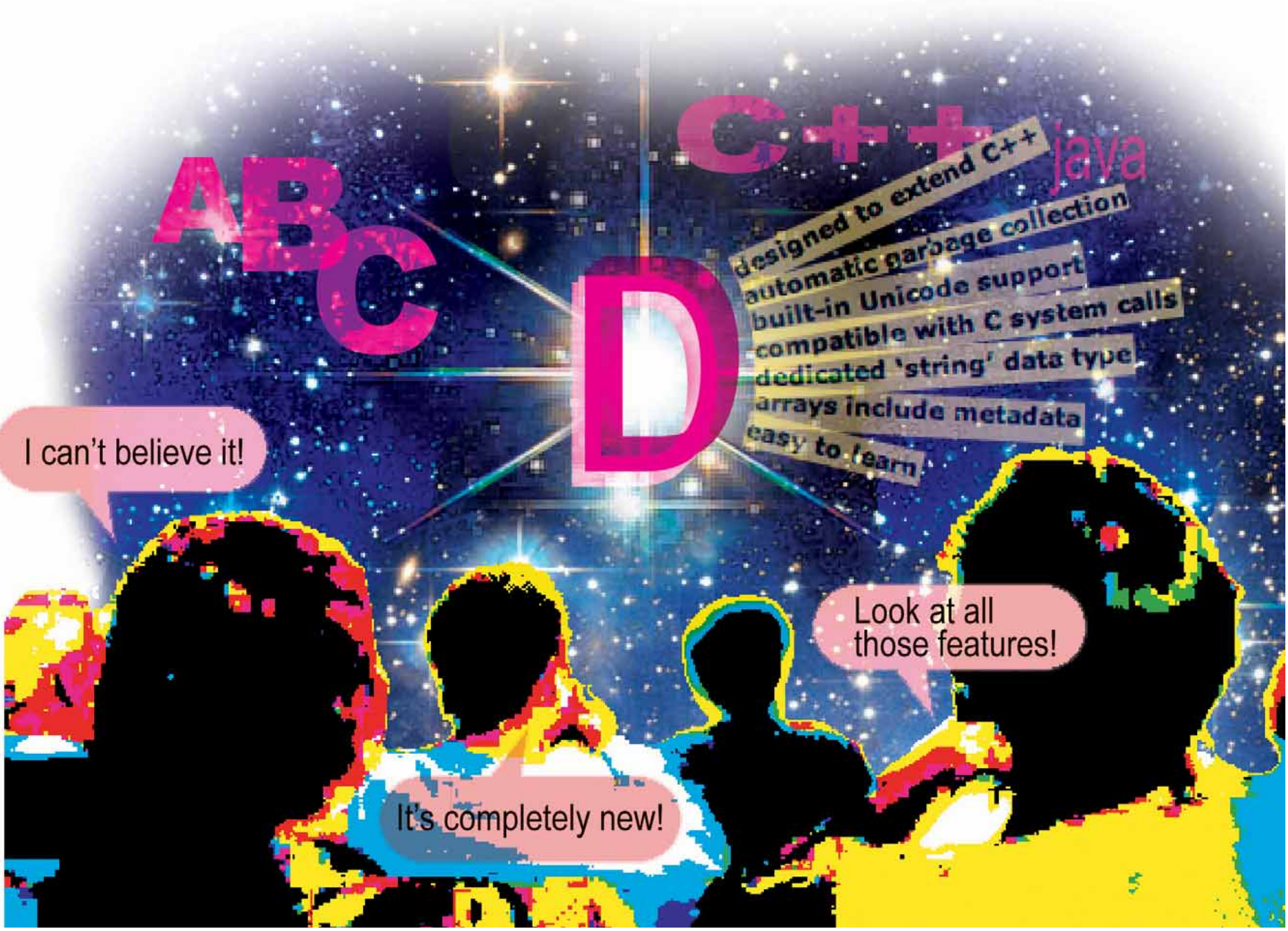
Если вы любите работать с подсветкой синтаксиса при редактировании кода, то вам будет приятно узнать, что многие редакторы уже поддерживают синтаксис D. *Vim*, начиная с версии 6.3, включает режим D как один из стандартных. Пользователи *Emacs* могут скачать модуль для D с **www.prowiki.org/wiki4d/wiki.cgi?EditorSupport/EmacsDMode**. Если вы используете Kate, то вам не повезло. Но даже если ваш любимый редактор не понимает D, вы можете попытать счастья с режимами Java или C#, поскольку их синтаксис наиболее близок к синтаксису D.

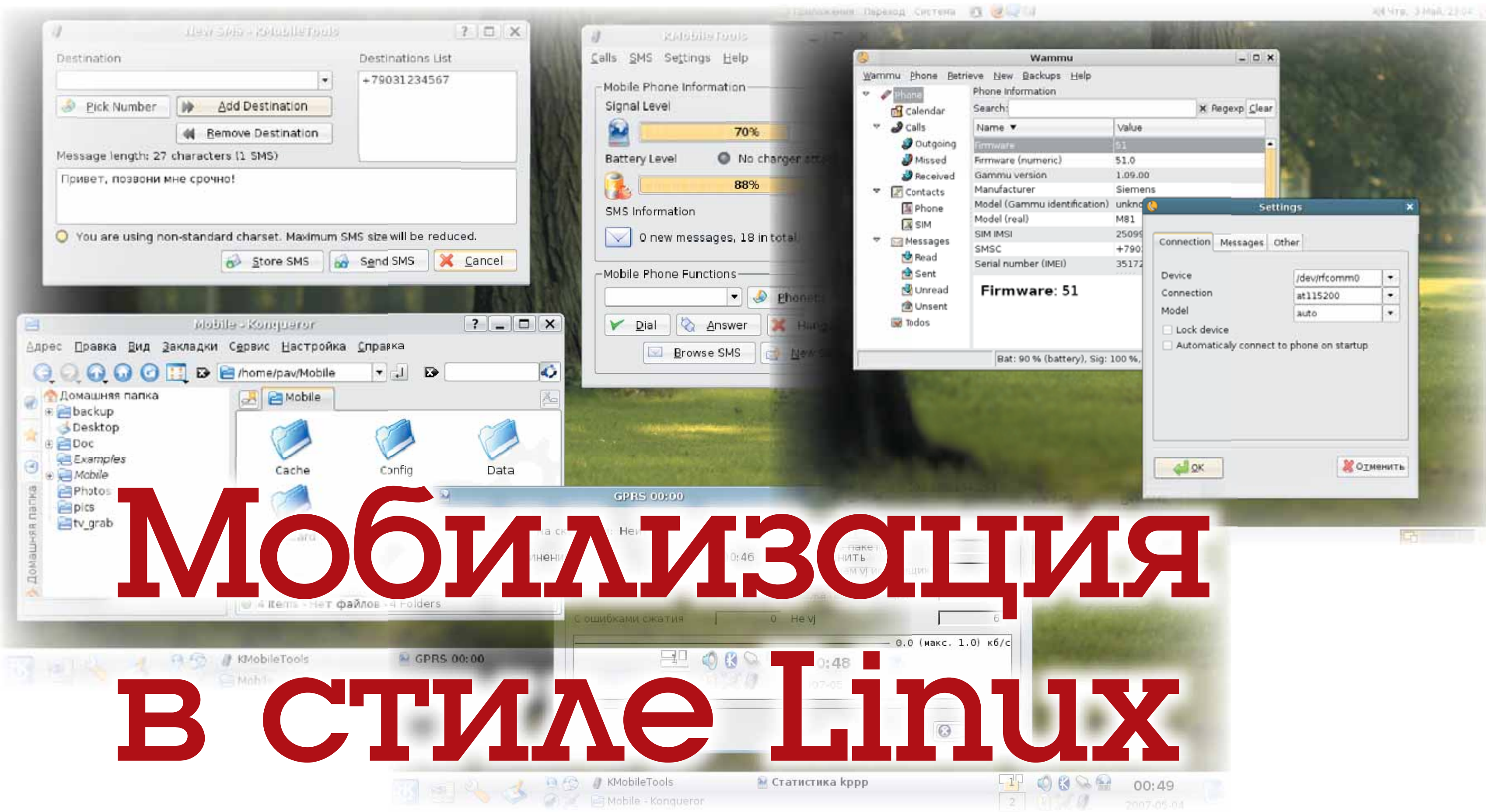
Круто! Большое спасибо за то, что рассказали мне про D. Если на этом все, то я пойду и попробую поработать с ним прямо сейчас!

Пожалуйста! А в следующем месяце мы с вами рассмотрим E.

E?! Это шутка, так ведь?

Подождите немного, и увидите. LXF





Мобилизация в стиле Linux

Навязчивая реклама утверждает, что мобильность – один из основных факторов, влияющих на имидж современного человека, и мы в редакции *LXF* ежедневно получаем пачки писем с просьбами «помочь настроить сотовый в Linux». К счастью, **Павел Воронцов** готов дать ответы на все интересующие вас вопросы.

Обладатели современных мобильных телефонов всегда сталкиваются с одной очень непростой задачей – обеспечением его связи с ПК или ноутбуком. Зачастую проблемы начинаются уже с момента покупки, когда радостный обладатель вполне многофункционального устройства обнаруживает отсутствие интерфейсного кабеля, а после его приобретения удивляется тому, что телефон при подключении не определяется как устройство хранения данных из-за отсутствия специализированного ПО. Конечно, есть другие интерфейсы взаимодействия, такие как IrDA и Bluetooth, но проблему установки ПО это не решает. Пользователям Windows все же проще – производители мобильных телефонов обычно не ленятся писать разнообразные, громоздкие и зачастую не слишком функциональные программы. А как быть тем, кто выбрал Linux? Попробуем разобраться.

OBEX. Обмен файлами

Для передачи данных между различными мобильными устройствами и ПК обычно используется протокол обмена объектами *OBEX* (OBject Exchange). На сегодняшний день, данный протокол реализован на многих устройствах: PDA, таких как Palm Pilot и мобильных телефонах Siemens, Nokia, Sony-Ericsson и других. *OBEX* был разработан организацией IrDA (Infrared Data Association) и является высокоуровневым протоколом. Хотя первоначально он создавался для инфракрасных сетей, сейчас его можно использовать и с другими транспортами, такими как TCP/IP и Bluetooth. В Linux этот протокол реализует открытый проект *OpenOBEX* (**http://openobex.triq.net**).

Для примера возьмем мобильный телефон Benq-Siemens M81 и попробуем настроить соединение в Ubuntu Linux 6.10. Данный телефон имеет два способа подключения к ПК: посредством кабеля или через Bluetooth. Далее, чтобы не путать читателя, мы будем пользоваться вторым из них, тем более, что он является более универсальным. В Linux существует несколько реализаций данного протокола, но мы остановимся на BlueZ (**www.bluez.org**). Соответствующий модуль должен быть проинсталлирован и загружен в ядро (в современных дистрибутивах это происходит автоматически), кроме того, потребуется установить два пакета: bluez-libs и bluez-utils. Последний содержит демоны Bluetooth, файлы конфигурации и консольные утилиты. Подключите Bluetooth-адаптер к ПК, внесите необходимые изменения в файл **/etc/bluetooth/hcid.conf** (см. ниже) и запустите демон:

```
$ sudo /etc/init.d/bluetooth start
```

В принципе, нам подойдут настройки по умолчанию. Рекомендую только проверить параметр security и, при необходимости, установить его значение в auto. Также стоит обратить внимание на параметр passkey. Это пароль, который будет автоматически передаваться при запросе соединения внешним устройством. Не поленитесь заглянуть в настройки Bluetooth на мобильном телефоне и проверить его: иногда проблемы связи возникают именно из-за этого. Я также добавил адрес своего Bluetooth-адаптера в список доверенных устройств.

С паролями все ясно, теперь давайте проверим, все ли мы правильно сделали. Запустите команду hciconfig (HCI – это Host Controller Interface или, по-простому, ваш Bluetooth-адаптер):

```
$ hciconfig

hci0:   Type: USB
        BD Address: 00:02:72:CC:EC:B2 ACL MTU: 1017:8 SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:391 acl:0 sco:0 events:17 errors:0
        TX bytes:317 acl:0 sco:0 commands:17 errors:0

```

Устройство определено и работает (статус: UP). Мы также видим его MAC-адрес: 00:02:72:CC:EC:B2. Если устройство имеет статус DOWN, попробуйте ввести команду:

```
$ sudo hciconfig hci0 up && hciconfig
```

Теперь давайте поищем наш мобильный телефон. Для этого служит команда:

```
$ hcitool scan

Scanning ...
        00:11:22:33:44:55       MyMobile
```

Как видно, устройство обнаружено: программа вывела имя и MAC-адрес телефона. Осталось только разобраться, как обеспечить обмен

файлами. *OpenOBEX* – это набор библиотек, реализующих протокол, а нам необходимы программы, использующие его. На сайте разработчика *OpenOBEX* вы найдете пакеты *ObexFTP* и *ObexFS* – они-то нам и понадобятся. Вот так, например, можно вывести содержимое корневого каталога нашего тестового телефона:

```
$ obexftp -b00:11:22:33:44:55 -B6 -l

Browsing 00:11:22:33:44:55 ...
Channel: 6
Connecting...done
Receiving "(null)"...\<?xml version="1.0"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version="1.0">
<folder name="Data" modified="19970101T000000" user-perm="W"
group-perm="W" />
<folder name="Cache" modified="19970101T000000" user-perm="W"
group-perm="W" />
<folder name="Config" modified="19970101T000000" user-perm="W"
group-perm="W" />
<folder name="MMCard" modified="19970101T000000" user-perm="W"
group-perm="RW" />
</folder-listing>
done
Disconnecting...done
```

Обратите внимание, что в качестве параметров *ObexFTP* я указал MAC-адрес телефона (00:11:22:33:44:55) и номер канала (6). Список всех доступных каналов определяется командой:

```
$ sdptool browse  00:11:22:33:44:55

...
Service Name: OBEX File Transfer
Service RecHandle: 0x11106
Service Class ID List:
  "OBEX File Transfer" (0x1106)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 6
  "OBEX" (0x0008)
Language Base Attr List:
  code_ISO639: 0x656e
  encoding:    0x6a
  base_offset: 0x100
Profile Descriptor List:
  "OBEX File Transfer" (0x1106)
    Version: 0x0100
...
```

Нас интересует профиль под названием «OBEX File Transfer», его номер канала (Channel) как раз и равен 6. Устройства Bluetooth пользуются каналами во многом аналогично портам TCP/IP: для передачи разнотипных данных с одного адреса. Вы можете, например, одновременно использовать стереогарнитуру, работать в Интернете и копировать файлы. Существует несколько профилей Bluetooth, которые, в зависимости от задачи, задействуют тот или иной канал связи. Если вы посмотрите на полный список, то увидите профиль под названием «SerialPort». С телефоном можно работать, как с обычным COM-портом, но для этого его нужно предварительно настроить утилитой *rfcomm*:

```
 $ sudo rfcomm bind rfcomm0 00:11:22:33:44:55 1
```

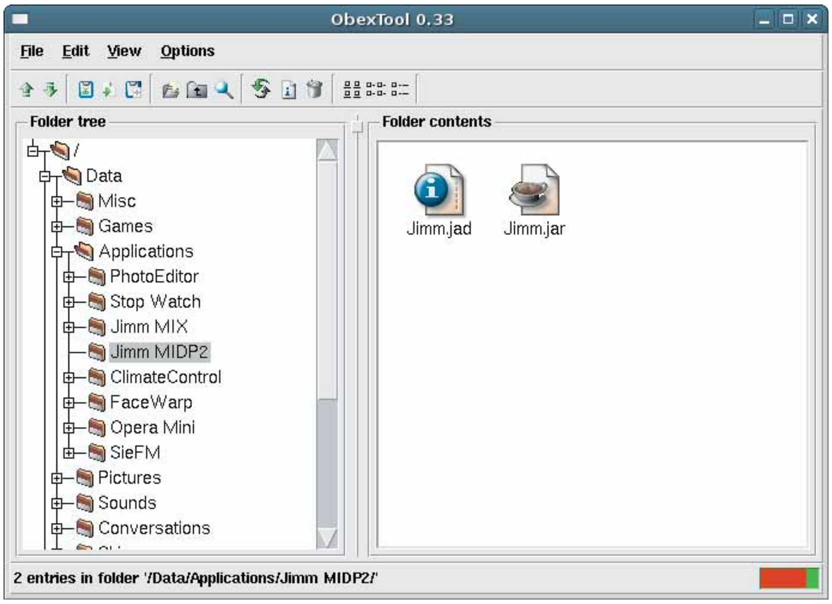
После этого вызов *ObexFTP* можно переписать, избавившись от малопонятных MAC-адресов и номеров каналов:

```
$ obexftp -t /dev/rfcomm0 -l
```

Если команда работает, то смело редактируйте конфигурационный файл **/etc/bluetooth/rfcomm.conf**, чтобы устройство создавалось автоматически при запуске демона bluetooth. Пример такого файла приведен ниже:

```
rfcomm0 {
        bind yes;
        device 00:11:22:33:44:55;
        channel     1;
        comment "Benq-Siemens M81";
}
```

Программа *ObexFTP* позволяет обмениваться файлами, но пользоваться ею не слишком удобно – это консольное приложение, выводящее данные в формате XML. Неудивительно, что для нее существует несколько графических оболочек. Из их числа стоить отметить *ObexTool* – это самое простое решение в нашей непростой задаче. Программа имеет простой графический интерфейс и позволяет исследовать файловую систему мобильного телефона, а также совершать основные действия с файлами и каталогами.



› **Рис. 1. Программа *ObexTool*.**

Есть и второй путь, более правильный и интересный: смонтировать файловую систему телефона как обычный диск. Проблема состоит в том, что вы не имеете доступа к файловой системе напрямую: все ваше общение происходит на уровне протокола *OBEX*, но выход есть – создание виртуальной файловой системы. В этих словах нет ничего страшного. Для нас, пользователей, это означает, что нужно установить еще несколько пакетов и произвести кое-какие настройки. Все остальное возьмет на себя FUSE (Filesystem in Userspace), неоднократно упоминавшийся на страницах нашего журнала. Это модуль ядра и библиотеки, которые позволяют просто создавать и монтировать пользовательские файловые системы в режиме пользователя. Модуль ядра, скорее всего, у вас уже установлен, так что необходимо просто добавить пакеты *libfuse2* и *fuse-utils* или собрать их самому. Все подробности можно найти на сайте **http://fuse.sourceforge.net**.

Для начала подготовим каталог для монтирования и выполним необходимые системные настройки. Во-первых, добавим текущего пользователя в группу fuse, чтобы он мог монтировать файловые системы:

```
$ sudo usermod -aG fuse имя_пользователя
```

Во-вторых, создадим каталог монтирования:

```
$ sudo mkdir -m777 /media/mobile
```

Теперь перейдем к тому, ради чего мы все это затевали: к монтированию виртуальной файловой системы *ObexFS*. Введем команду:

```
$ obexfs -b00:11:22:33:44:55 -B6 /media/mobile
```

или

```
$ obexfs -t /dev/rfcomm0 /media/mobile
```

Файловая система будет смонтирована в **/media/mobile**. Теперь вы можете пользоваться вашим любимым файловым менеджером, и »

Скорая помощь

Хочется также отметить еще одну виртуальную файловую систему, SieFS (http://chaos.allsiemens.com/siefs). На странице автора сказано, что она предназначена для мобильных телефонов фирмы Siemens. Работоспособность на телефонах других фирм я не проверял.

работать с файлами мобильного телефона как с обычным носителем. Для размонтирования необходимо ввести команду:

```
$ fusermount -u /media/mobile
```

Существуют и другие способы монтирования, например, всем известной командой *mount*, также есть возможность прописать файловую систему в **/etc/fstab**. Все эти способы подробно описаны на странице программы.

Hayes. Совсем не забытое старое

Итак, с протоколом *OBEX* мы немного разобрались. Обмен файлами – задача важная, но не единственная. Плюс мобильного телефона – маленький размер – одновременно является и его минусом. ПК может сделать работу с телефоном более комфортной: например, читать SMS-сообщения с экрана монитора ПК намного приятнее, чем с маленького дисплея, а набирать их на большой клавиатуре куда удобнее, чем на маленькой 12-клавишной. Здесь возникает целый спектр задач: управление телефонными книгами, резервное копирование данных, синхронизация, да и просто использование коммуникационных возможностей мобильного телефона с ПК.

Для управления телефоном используется протокол *Hayes*. Это очень старый протокол, который был создан для управления модемами, но не потерял актуальности и по сей день. Он поддерживается практически всеми мобильными телефонами. Управление происходит посредством так называемых AT-команд. Есть стандартные команды, которые подойдут к любому устройству, но есть и специальные для конкретной модели телефона. К счастью, нам не придется писать эти команды вручную, поскольку существует множество разнообразных программ, способных сделать это за нас.

Начнем, как это ни странно, с консольных приложений. Преимущества таких программ очевидны: они очень мощные и незаменимы при автоматизации вашей работы. Например, администратор, желающий получать оперативную информацию о работе сервера каждый час или при возникновении аварии, с легкостью напишет скрипт, использующий консольную программу, которая будет отсылать SMS с нужной информацией на определенный номер. Самый простой вариант – программа *SCMxx* (http://www.hendrik-sattler.de/scmxx). К сожалению, она ориентирована на мобильные телефоны Siemens. Во всяком случае, автор не привел в списке совместимых устройств телефоны других фирм, но, скорее всего, большинство функций должно работать. Вот простой пример ее использования: данная команда отправляет Flash-SMS на заданный номер:

```
$ scmxx --device=/dev/rfcomm0 --send --sms --direct --flash --unicode --number="+79031234567" --text="Привет, я flash-sms!"
```

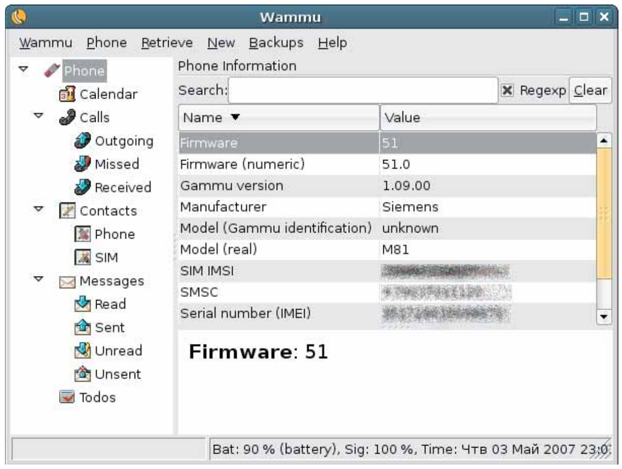
Flash-SMS – это специальное сообщение, сразу же отображаемое на экране телефона при получении (в разных аппаратах это реализовано по-разному). Такие сообщения обычно присылают сотовые операторы и с обычного мобильного телефона отправить его нельзя. Но так как программа сама формирует SMS-сообщение и может выставить любые значения в его служебных полях, то вы без труда отправите такое SMS, поразив тем самым своего друга или подругу. Разумеется, это не все, на что способна программа; вот пример резервного сохранения телефонной книги телефона в файл:

```
$ scmxx –device /dev/rfcomm0 --get --pbook --mem=SM --out=SM.pb
```

Обратимся к другим вариантам. *Gammu* (http://www.gammu.org) – это даже не программа, это целый проект, который активно развивается. Список поддерживаемых устройств очень велик, функциональность просто впечатляет. *Gammu* состоит из двух частей: ядра, представленного библиотекой, и самой консольной программы *gammu*. Такое исполнение очень удобно для разработчиков, которые не заставили себя долго ждать: на данный момент существует порядка 10 разных графических оболочек, использующих ядро *gammu*. Перейдем сразу к обзору основных графических оболочек: они, конечно, не обеспечивают полный функционал ядра, но с ними легче работать пользователю.

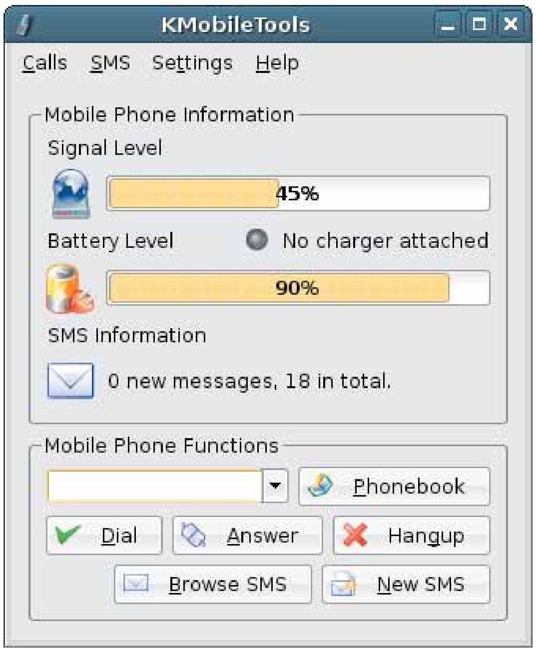
Для начала установим пакет *libgammu1*. Он содержит то самое ядро, которое нам так необходимо. Пакет *gammu* содержит консольную программу, и если вы не хотите пользоваться ею, устанавливать его не обязательно.

Начнем с программы *Wammu* (http://wammu.eu). Это простой менеджер мобильного телефона, позволяющий редактировать телефонную книгу, читать и отправлять SMS-сообщения, звонить, делать резервные копии. *Wammu* также умеет работать с календарем и заметками. Программа написана на *wxPython* и использует *wxWidgets*, кросс-платформенный GUI-инструментарий [в Linux его вызовы обычно транслируются GTK+, – прим. ред.]. Это означает как то, что программа может работать и в других операционных системах, так и то, что вам придется установить соответствующие библиотеки.



› **Рис. 2. Главное окно программы *Wammu*.**

Вторая программа – *KMobileTools* (http://www.kmobiletools.org). По функциональности очень похожа на предыдущую, но гораздо удобнее и обладает очень приятным интерфейсом. Программа написана для KDE, соответственно использует KDE-библиотеки. *KMobileTools* не работает с органайзером, зато может синхронизировать адресную книгу мобильного телефона с адресной книгой KDE.



› **Рис. 3. *KMobileTools* собственной персоной.**

Существуют и другие программы, ориентированные на конкретные модели телефонов. Например, для аппаратов Nokia существует программа *gnokii* (http://www.gnokii.org) со своими графическими оболочками, а для телефонов Motorola – программа *moto4lin* (http://moto4lin.sourceforge.net). Также, не помешает заглянуть на сайт производителя

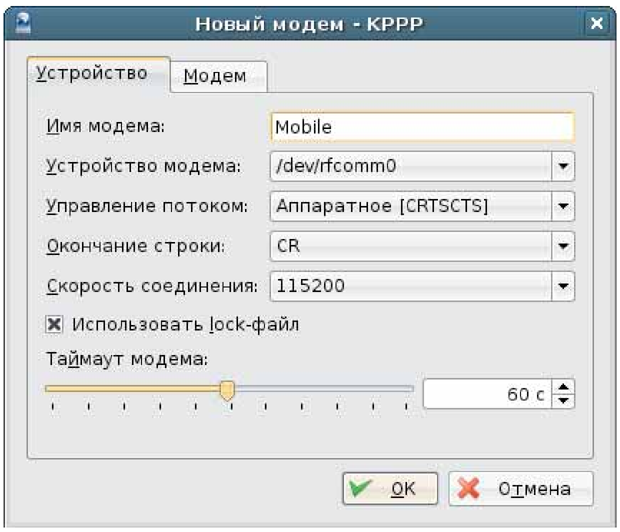
телефона, там тоже можно найти специализированное ПО для Linux. Например, для телефонов Benq-Siemens существует программа *XMPM* (*X Mobile Phone Manager*).

GPRS – это просто

Перейдем к заключительной части статьи: настройке подключения к Интернету через GPRS. GPRS (General Packet Radio Service) – это надстройка над технологией мобильной связи GSM, осуществляющая пакетную передачу данных. Это позволяет обмениваться данными с внешними сетями, в том числе, Интернетом. При использовании GPRS информация собирается в пакеты, которые передаются через неиспользуемые в данный момент голосовые каналы. Такая технология предполагает более эффективное потребление ресурсов сети GSM. Существуют и другие технологии передачи данных, например CSD (Circuit Switched Data). По сути, это аналоговый модем, работающий в голосовом канале сети GSM, но ее скорость и качество связи явно уступают GPRS. С учетом того, что платить абоненту придется за время, а не за трафик, использовать CSD просто невыгодно.

Вместо того, чтобы запутывать читателя «рецептами» настройки соединения с Интернетом через GPRS в Linux, мы постараемся объяснить сам принцип этой настройки. В последний раз я настроил GPRS-соединение спустя минуту после подключения телефона к ПК, и не потому что я фокусник, а потому что это действительно просто.

Соединение устанавливается посредством протокола PPP (Point-to-Point Protocol) – это стандартный механизм для создания и запуска IP (Internet Protocol) и других сетевых протоколов на последовательных линиях связи. Для наглядности процесса будем использовать программу KPPP.



› **Рис. 4. Создать новый модем в KPPP несложно – но лучше, чтобы скорость не превышала 115200.**

Для начала добавим и настроим модем. Тут надо выбрать устройство последовательного порта, в нашем случае это будет **/dev/rfcomm0**. Необходимо также написать дополнительную строку инициализации для создания профиля соединения. Она имеет следующий вид

```
AT+CGDCONT=<cid>,<PDP_type>,<APN>
```

<cid> – это идентификационный номер создаваемого профиля, именно на него будет ссылаться команда при установке соединения. Обычно этот параметр равен «1», но вы можете использовать и другой номер, например, для создания второго профиля для другого оператора. <PDP type> – тип соединения – “IP”, <APN> – адрес точки доступа, его нужно узнать у оператора. Для уверенности попробуйте ввести эту команду в терминальной программе (скажем, *CuteCom*, см. LXF92) и посмотрите на ответ телефона: он должен быть положительным. Например, для оператора Beeline строка инициализации будет следующей: AT+CGDCONT=1,”IP”,”internet.beeline.ru”.

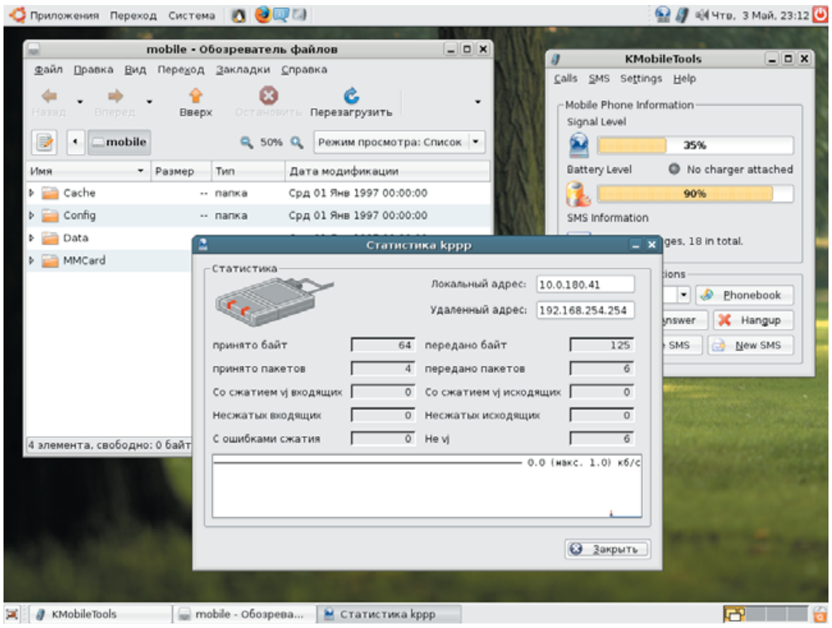
Теперь необходимо ввести номер телефона. По сути, это вовсе не номер, а продолжение команды ATD, которая должна установить GPRS-соединение для созданного нами профиля. Вот ее формат:

```
ATD*<GPRS_SC>*<L2P>*<cid>#
```

Параметры разделены символами * и заканчиваются символом #. <GPRS_SC> – это идентификатор принадлежности строки к GPRS, а не к аналоговому соединению, он всегда равен 99 (GPRS Service Code = 99). <L2P> – требуемый протокол 2-го уровня, в данном случае, значение параметра можно опустить. И наконец, <cid> – идентификатор профиля, созданного при инициализации. Как видите, все просто. «Номер телефона» – *99***1#. Осталось узнать у оператора имя пользователя и пароль и установить соединение.

В принципе, на этом можно было бы и закончить, но есть еще один маленький нюанс. Мы настроили файловую систему, менеджер мобильного телефона и подключение к Интернету, используя одно и то же устройство – **/dev/rfcomm0**. Это нерационально, так как вы не сможете использовать все услуги одновременно: постоянно будут конфликты. Необходимо настроить разные задачи на собственные каналы Bluetooth. Так как протокол *OBEX* менее требователен к типу подключения, для него не обязательно выделять COM-устройство (**/dev/rfcomm0**). Лучше подключиться напрямую, указывая MAC-адрес и номер канала связи (6). С протоколом *Hayes* сложнее, ему необходима эмуляция COM-порта. Среди профилей Bluetooth вы найдете один с именем «Dialup Networking». В моем случае, его номер канала равен 2. Создайте еще одно устройство – **/dev/rfcomm1**, настройте его на 2-й канал и используйте для подключения к Интернету:

```
$sudo rfcomm bind rfcomm1 11:22:33:44:55 2
```



› **Рис. 5. Различные программы не мешают друг другу, потому что используют различные каналы Bluetooth.**

Думаю, мне удалось показать, что работать с мобильным телефоном в Linux можно ничуть не хуже, чем в Windows, более того: в Linux эта работа эффективнее, а иногда и проще. Здесь есть очень много универсальных программ, а в Windows их меньше, из-за наличия родного ПО от производителя. В том, что касается виртуальной файловой системы, Linux вообще вне конкуренции. Есть, конечно, и минусы. Самый яркий – отсутствие программ для обновления прошивки телефона и работы с Flash-памятью, но, надеюсь, это всего лишь вопрос времени. LXF

Свежайший пакет Sun Studio 12 уже доступен всем!

Sun Studio 12 – это новый свободно распространяемый интегрированный пакет средств разработки для Solaris и Linux.

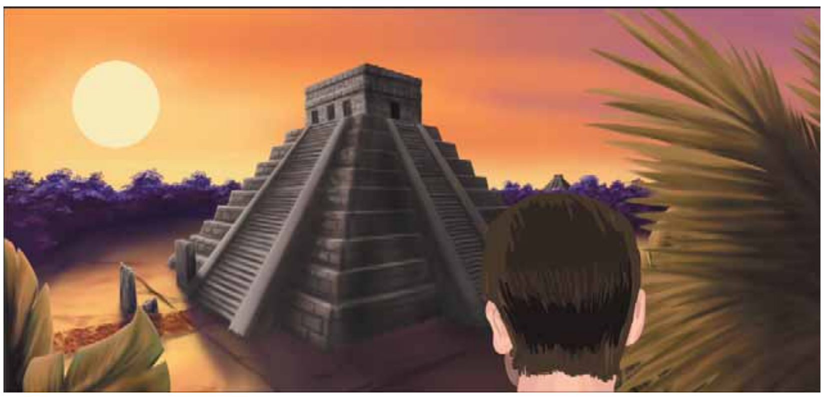
Важная новинка **Sun Studio 12** - совершенно новая среда разработки.
Разрабатывать программы стало удобнее: новая графическая среда позволяет концентрироваться на работе, инструменты использовать проще. Кроме того, значительно улучшена поддержка Linux: теперь **Sun Studio** под Linux имеет столь же богатую функциональность, сколь и под Solaris.
Все компиляторы **Sun Studio 12** поддерживают OpenMP и позволяют автоматически распараллеливать программы.

Sun Studio 12 поможет вам выиграть до **$5000** в игре **Temple of the Sun** (www.sun.com/templeofthesun).
Принять участие в игре можно до **19 июня 2007 года**.
Все подробности – по адресу:



http://ru.sun.com/developers/sunstudio/articles/templeofthesun.html

Скачать **Sun Studio** бесплатно можно по адресу
http://ru.sun.com/developers/sunstudio/
Также вы можете оставить заявку на странице http://ru.sun.com/developers/sxde/request/
и бесплатно получить DVD с дистрибутивами Solaris Express Developer Edition и SunStudio.







Учебники

Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Третья стадия?

«Сначала они тебя не замечают, потом смеются над тобой, затем борются с тобой. А потом ты побеждаешь.»

Мохандас Карамчанд «Махатма» Ганди.

Это отнюдь не новость. Там (махая рукой в предположительно западном направлении) борются уже давно и на полном серьёзе. Но у нас (позиционируя себя на участке площадью примерно в 12% от всей суши) это время только-только наступает. Те, кто это делает публично через средства массовой информации, ещё даже не выучили правильные слова и заклинания, поэтому прямо сейчас появляются перлы вида «Корпорация Linux».

Вряд ли одна очень богатая фирма специально инициировала судебный процесс против отдельно взятого директора отдельно взятой школы, но она создала/создаёт/будет создавать все условия для их возникновения.

GNU/Linux в России — это уже данность, хотя бы потому, что Министерство финансов может отказать в выделении из федерального бюджета средств на легализацию программного обеспечения в российских школах – проект, курируемый самим Дмитрием Медведевым, ссылаясь на эффективность открытых систем. И в этом ему помогли сами авторы проекта с весьма странным подходом «закупить для всех школ *Фотошоп*, потому что его школы просили». Если Вы ещё не в курсе того что происходит, то рекомендую прочитать «Мининформсвязи: положение с лицензированием школьного ПО в России угрожающее» Александра Костинского (http://www.svobodanews.ru, *набрать в поиске «Костинский лицензирование»*) или хотя бы наши новости в начале журнала.

В конце хотелось бы отметить на редкость грамотную газетную статью Юрия Ревича «Встряхнуться перед потреблением» (http://novayagazeta.ru/, *в поиске набрать «Юрий Ревич Linux»*). Побольше бы таких статей.

E.M.Baldin@inp.nsk.su

В этом выпуске...



58 Знакомство с файловой системой
Новичок в Linux? Затрудняетесь расшифровать все эти usr и var? Не беспокойтесь – **Энди Ченнел** вам все расскажет...



70 ООП: классы
В прошлый раз вы познакомились с объектно-ориентированным программированием – а сегодня **Пол Хадсон** покажет вам магию наследования.



62 Устанавливаем VPN
Открытое приложение Hamachi позволяет создать виртуальную частную сеть без лишних телодвижений. **Рейчел Проберт** объясняет, как пользоваться ее GNOME-оболочкой, GHamachi.



74 В духе Blender
Это не только инструмент 3D-моделирования, это еще и платформа для... чего угодно! **Бен Харлинг**, например, программирует в нем чат-приложения.

66 Программирование для web
Майк Сондерс вознамерился научить мир... как запрограммировать интернет. Сегодня мы возьмем Nvu и Base, свяжем их клеем из PHP – и получим прекрасный менеджер коллекций, доступный через браузер!



78 Блокнот, подвинься!
Андрей Боровский не похож на пересмешника. Все, чего мы хотим – это поделиться удачными приемами использования Anjuta и рассказать о паре тонкостей GTK+...



82 JDBC: ушел на базу
Александр Бабаев готов поведать о том, что на самом деле происходило за кулисами интерфейса OOo Base в учебнике Майка Сондерса...



86 Все о таблицах
Хвосты, головки, боковики – **Евгений Балдин** препарирует таблицы с ловкостью профессионального хирурга и показывает, как облечь их в LaTeX.



90 Подписи цифрой
Андрей Боровский объясняет, с чем едят электронные сертификаты и что это вообще такое...



94 Композитинг
Продолжаем изучать нелинейный видеоредактор *Cinelerra* с **Петром Семилетовым**.
Внимание: не усвоившие этот урок к последующим занятиям не допускаются!



97 Ужимаем видео
Перегнать DVD Video в обычный AVI? Нет ничего проще – благодаря MEncoder и учебнику **Евгения Гаврилова**!



Совет месяца: Тайм-менеджмент



Как часто вашим системам удавалось успешно перейти с зимнего времени на летнее? Это важный момент, поскольку часы – это не просто еще один апплет в панели задач, это механизм, влияющий на работу компьютера, как никакой другой. Будь то поле в заголовке отправленного email-сообщения или отметка времени на файле, все в вашей системе зависит от часов.

Простейший способ проверить точность их хода – использовать команду *date*. Ее вывод – одна-единственная строка, содержащая сведения о дате и времени в специальном формате. Для их установки с помощью *date* используется тот же синтаксис, хотя его можно легко настроить. Многочисленные параметры командной строки позволяют изменять точность от числа наносекунд до текущего века.

Последнее поле в выводе date показывает часовой пояс, на который настроена ваша система. Например, в Великобритании в настоящее время года им, вероятно, будет BST (British Summer Time, Британское летнее время). Пояс задается в файле **/etc/timezone**, который содержит описание вашего текущего местоположения. Для BST там, скорее всего, содержится Europe/London. Если настройка окажется некорректной, вы можете подобрать более подходящую из каталога **/usr/share/zoneinfo/**. В нем содержится большое число мест, достойных вашего проживания, отсортированных по континентам и странам.

В вашей системе присутствуют двое часов. Это системные часы (на них-то и воздействует программа *date*), а также аппаратные часы. Последние находятся рядом с BIOS и продолжают «тикать», даже когда компьютер выключен. Системные часы устанавливаются по аппаратным в процессе загрузки. Для опроса и перевода последних существует команда *hwclock*. Так, набрав и выполнив

```
hwclock --systohc
```

вы, наоборот, установите аппаратные часы по системным.

Система:

Не спотыкайтесь о вашу файловую систему: порхайте по **/usr**, **/opt** и **/dev** с грацией бывалого линуксоида. **Энди Ченнел** объяснит, как она работает.





Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32

Одно из первых препятствий, на которое натыкается новичок в Linux, это загадочная и пугающая файловая система. Где, к примеру, ободряющий значок Мой компьютер? Что такое **/usr**? Где **C:**, **A:** и прочие привычные обозначения используемых устройств? К счастью, как будет показано в одиссее ОС этого месяца, файловая система Linux сложна лишь на вид – и большую ее часть можно без проблем проигнорировать, если только вы не собираетесь стать хакером Linux или системным администратором.

Составить представление о структуре вашей компьютерной системы вполне возможно, и это определенно стоит сделать. Во-первых, знакомство с файловой системой поможет вам избежать случайных повреждений (т.е. удаления жизненно важного для системы файла без возможности восстановления), да неплохо и знать, где искать, если загруженный файл или иконка приложения не появились в ожидаемом месте.

Так что приглашаю в путешествие по файловой системе Linux. Файловая система просто описывает, как ваш компьютер организует различные папки и файлы, среди которых присутствуют операционная система, приложения и данные – вроде документов, картинок и музыки. На данном уроке я покажу вам, что делают элементы системы, сосредоточившись на основных (с точки зрения пользователя), и отмечу, в чем их раскладка сходна или отлична от Windows-аналогов.

Предупреждаю, что хотя мы будет исследовать данную структуру преимущественно как пользователь, иногда нам потребуется «стать супером» для полного исследования каталога. В таких случаях следует позаботиться о том, чтобы случайно не изменить жизненно важные файлы: результатом может быть гибель установленной системы. Для получения привилегий суперпользователя откройте терминал и наберите su, затем нажмите Enter и введите пароль root. Затем вы можете открыть *Nautilus* или *Konqueror*, для Gnome и KDE соответственно, из командной строки, введя nautilus или konqueror.

Держа это предупреждение в голове, незамедлительно начнем сверху... или снизу... ну вот, я сам запутался!

Причуды Gnome

Для начала откройте ваш файловый менеджер (обычно это будет *Konqueror* или *Nautilus*, но большинство дистрибутивов обозначают их просто как Файловый менеджер). В Ubuntu, Fedora или любом другом дистрибутиве на базе Gnome вам необходимо кое-что поменять, прежде чем мы продолжим, поскольку последние версии Gnome предпочитают прятать древо файловой системы.

Для легкой перестройки Gnome, выберите меню Переход, а затем Домашняя папка. Далее я буду обозначать подобные операции так: Переход > Домашняя папка. Стандартный способ работы в Gnome предусматривает, что двойной щелчок на каталоге открывает новое окно. Это хорошо при мониторе размером с киноэкран и организаторских способностях, как у колонии муравьев, но большинство из нас очень скоро увязнет. Поэтому выполните Правка > Параметры и выберите вкладку Поведение. Щелкните на флажке с надписью Всегда открывать в окне браузера, а затем нажмите кнопку Закрыть. Теперь закройте браузер файлов и вновь выполните Переход > Домашняя папка. На сей раз менеджер файлов отобразит файлы в главной части окна. По верхнему краю отобразится несколько иконок браузера – Назад, Вперед, Вверх, Обновить и т.д. – а внизу слева появится список различных мест, куда можно перейти. Под панелью инструментов будет так называемая «тропка из хлебных крошек» (помните Мальчика-с-пальчик?). Она отображает маршрут к вашему текущему положению в файловой системе через серию кнопок, на которые можно нажимать. Чтобы переключиться к текстовому отображению пути, нажмите кнопку с иконкой записной книжки у левого края.

Теперь, после настройки, можно двигаться далее. Дважды щелкните на иконке панели Переход с названием Файловая система. Сразу же заметим, что файловая система Linux имеет единую иерархию. Дискеты, съемные диски и CD, вставляемые в компьютер, становятся частью этой иерархии. Она называется унифицированной файловой системой – в отличие от мира Windows, где эти устройства считаются отдельными, или разделенными, местами в файловой системе.

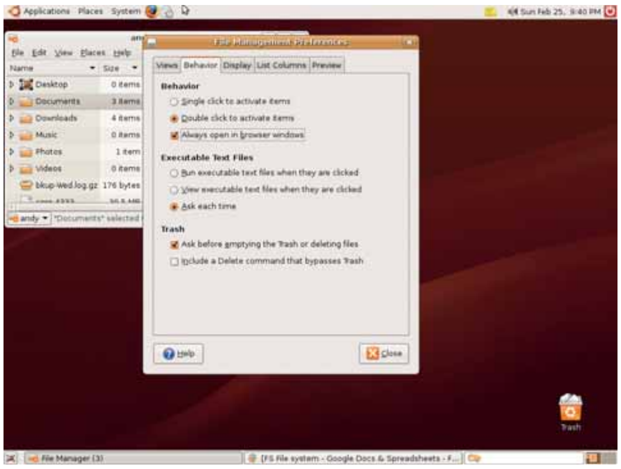
Наверху этой иерархии – то есть он содержит все остальные элементы – находится **/**, так называемый слэш- или корневой каталог (не путайте с **/root**). Когда мы щелкали на иконке Файловая система, первым появился **/**. Ниже располагается ряд каталогов (в Ubuntu их 19), хранящих другие каталоги и файлы. Принято, чтобы **/** содержал только каталоги, но некоторые дистрибутивы размещают в нем пару файлов. Вы не делайте этого!

/home

Для большинства из нас **/home** – самый важный каталог в системе Linux, потому что в нем располагаются домашние папки всех пользователей,

» **Месяц назад** Я научил вас процедуре безболезненного резервирования.

Иерархия каталогов



› *Nautilus* **работает лучше, если его настроить по типу браузера – ну, я так думаю! Это легко изменить в Параметрах.**

имеющихся в системе. Вероятно, там находится все: фотографии, музыка, презентации и документы. Здесь также располагаются файлы настроек конкретного пользователя, вроде тех, что определяют фон рабочего стола, оконные рамки и параметры приложений и данные, связанные с жизнью в сети – история браузера, архив электронных писем и так далее.

Каталоги в **/home** обычно доступны только пользователю, которому принадлежат, и суперпользователю (root, или системному администратору). При первом взгляде на **/home/username** вы можете не увидеть никаких файлов конфигурации. Дело в том, что они скрыты; но вы можете их «засветить», перейдя в меню Вид и выбрав Скрытые файлы. Тогда окно заполнится гуще, и вы отметите, что все ранее скрытые элементы начинаются с точки (**.**). Это способ Unix/Linux прятать файлы и каталоги от пользователя, хотя если вы дважды щелкнете по любому из них, они откроются так же, как и обычный.

Однако будьте осторожны: не трогайте скрытые файлы и папки, если не ведаете, что творите. Часто они содержат важные настройки, и любое некорректное изменение или удаление может серьезно затруднить возможность использования конкретного приложения.

Здесь есть каталог **Desktop**; перенесите что-нибудь в него, и оно возникнет на рабочем столе пользователя. Windows-эквивалентом этой папки является **C:\Documents and Settings\имя_пользователя\Desktop**.

/root

Каждый пользователь имеет собственный каталог, и пользователь root – не исключение. Его каталог аналогичен каталогу обычного пользователя и содержит документы и каталог **Desktop**. Однако он не должен быть особо заполнен документами, потому что рутинные работы пользователь root не выполняет, а значит, в этой учетной записи данным накапливаться не с чего.

/bin

Двигаемся дальше. **/bin** – это сокращение от binaries [двоичные файлы], и здесь вы найдете набор исполняемых файлов с загадочными именами вроде **ls**, **rm**, **mkdir** и **rmdir**, которые весьма полезны для системных администраторов и продвинутых пользователей. Если вы немного знакомы с командной строкой Linux, то уже сталкивались с этими именами, а те, кто работал только с графическими инструментами, пользовались этими программами, не подозревая об этом. Давайте рассмотрим некоторые их функции.

Откройте терминал, который обычно находится в меню под именем Терминал или Консоль. Когда он появится на экране, наберите

```
mkdir LXF
```

В терминале ничего не произойдет, но, открыв менеджер файлов и заглянув в ваш домашний каталог, вы должны увидеть новую папку с именем **LXF**. Она здесь потому, что терминал по умолчанию открывается в домашнем каталоге, а mkdir – это команда “make a directory” [создать каталог] (то же происходит, если вы щелкаете правой кнопкой мыши в окне и выбираете Создать каталог). Теперь в терминале наберите

```
rmdir LXF
```

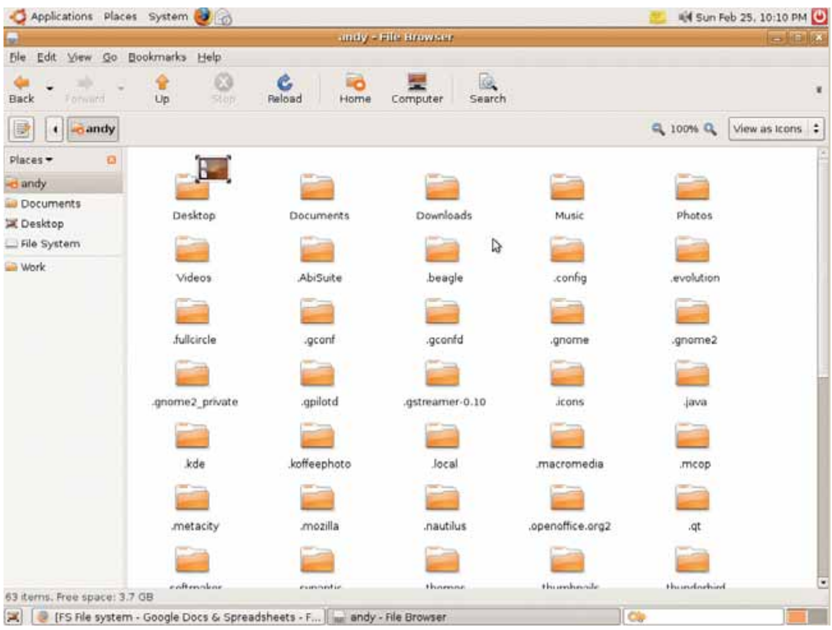
Новый каталог будет удален – как если бы вы после правого на нем выбрали пункт Удалить. Кстати, если вы не хотите ради проверки запускать менеджер файлов, просто наберите ls, и увидите содержимое каталога.

/sbin

/bin содержит приложения, необходимые вам для ежедневной работы (пусть вы и не осознаете этого); а **/sbin** – место, где система Linux хранит элементы, запускаемые в фоновом режиме [а точнее, инструменты, предназначенные для администратора, – прим.ред.]. Windows-эквивалентом ему может быть каталог **C:\Windows\System**. Здесь вы можете найти такие программы, как *Lilo* – загрузчик Linux, *fdisk* – для работы с различными разделами диска, и все остальное ПО, помогающее в работе системы. Нечего и говорить – но я все-таки скажу – что баловство с содержимым этого каталога способно серьезно повредить вашу ОС, поскольку оно жизненно важно для каждого пользователя.

/usr

Этот каталог немного странноват, поскольку содержит множество вещей, которым самое бы место прятаться где-то в **/bin** или **/sbin**. »



› **Файлами можно управлять из командной строки, но графический менеджер чуть проще.**

» Отличие в том, что эти двоичные файлы более приближены к пользователю. В Windows вы можете видеть подобное в **C:\Windows\Program Files** (хотя следует помнить, что Windows, в отличие от Linux, не чувствителен к регистру), и если вы хотите вручную запустить приложение, установленное при помощи менеджера пакетов, но не появившееся в меню, то начинать нужно отсюда.

Внутри **/usr** много подкаталогов. Их количество и имена зависят от используемого дистрибутива, но всегда имеется **/bin** для бинарных файлов, **lib** для файлов библиотек и **X11**, где вы найдете все для взаимодействия вашего компьютера со внешним миром. Здесь же, внутри каталога **/src**, располагаются исходные тексты ядра Linux. Можете посмотреть их, но они не очень-то интересны.

Так же как в **/bin** и **/sbin**, хорошей идеей будет перемещаться по этим каталогам осторожно – здесь легко учинить катастрофу [напомним еще раз: проблемы могут возникнуть, только если вы работаете от имени root. Однако, для простого просмотра системных каталогов привилегии суперпользователя обычно не нужны, а обычный пользователь при всем желании не сможет нанести вред **/usr** и другим системным каталогам, – прим. ред.].

/opt

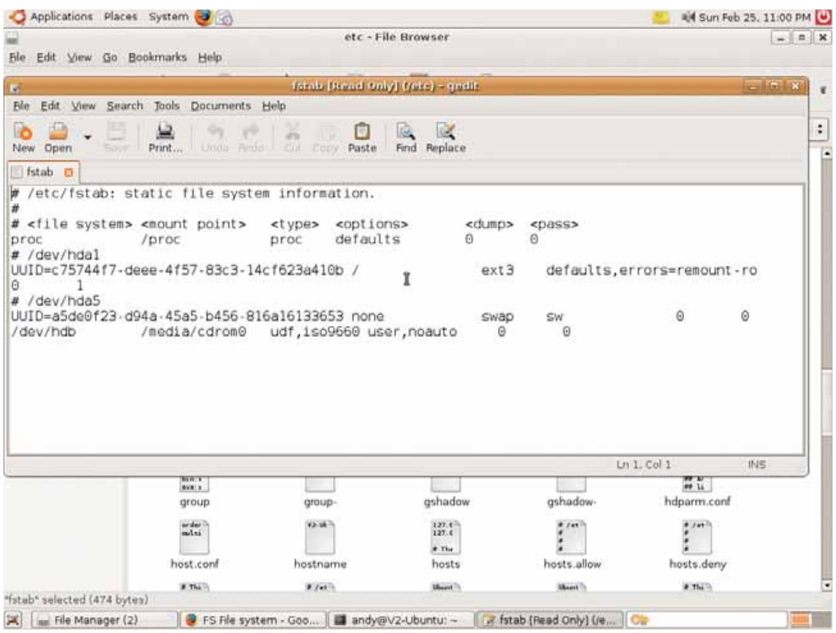
В стандартной структуре Unix, многие приложения, не являющиеся частью основного дистрибутива, помещаются в **/opt**, но в Linux он используется не часто. Если вы тщетно искали двоичный файл приложения в различных подкаталогах **/usr**, то может быть, он расположен где-то здесь [в некоторых дистрибутивах в **/opt** устанавливаются крупные приложения, такие как KDE, Gnome и *OpenOffice.org*. Обычно для них создается каталог с подходящим именем, внутренняя структура которого повторяет файловую систему Unix «в миниатюре»: здесь можно найти те же **bin**, **lib** и т.п., – прим. ред.].

/etc

Те же предупреждения верны и для каталога **/etc** – он содержит многие файлы конфигурации, отвечающие за корректную работу вашего компьютера. Например, здесь находится файл с именем **fstab**, разъясняющий Linux, как обращаться с различными монтируемыми файловыми системами. В системах на основе Debian в **/etc/apt** имеется файл **sources.list**, точно описывающий, откуда системе можно брать новые программы. Одним из наиболее важных подкаталогов здесь – и его регулярно нужно резервировать (см. предыдущий номер) – это **rc.d**; он содержит стартовые скрипты системы. Заглянув в каталог, вы увидите, что там больше одного **rc.d**. На самом деле их семь: **rc0.d**, **rc1.d**, **rc2.d** и так далее. Они соответствуют различным уровням запуска [runlevel], используемым в Linux.

Что такое уровень запуска? Он определяет количество запускаемых программ: будут ли это текстовые службы или полный графический интерфейс. Например, уровень запуска 1 обеспечивает простейшую реализацию системы (она полезна для поиска неисправностей, но вам вряд ли понравится), тогда как уровень запуска 5 – это полностью графическая загрузка с поддержкой сети, которую вы видите каждый день. Уровень запуска 0 – это выключение [halt], а 6 – перезагрузка. Пожалуй, с уровнями запуска связываться не стоит, но если, допустим, вы захотите перезагрузить машину из командной строки (от имени root), то можете просто набрать:

```
init 6
```



› *Tomboy* создает ссылки еще до того, как вы завершите их ввод… Умен!

О Боже… кругом файлы…

Во время нашего прочесывания файловой системы вы могли отметить, что все рассмотренное нами является файлом. Часть системы, отвечающая за перевод перемещений мыши в движение курсора на экране – это файл, и нечто, информирующее компьютер, сколько пикселей и цветов отображает ваш монитор – тоже… файл.

Это верно и для Windows, но в системе Linux эти файлы более доступны и полностью поддаются редактированию. В прошлом, большая часть настройки системы состояла в поиске в каком-то Богом забытом каталоге текстового файла с расширением **.conf** и в последующей его правке от имени пользователя root. Не удивительно, что это казалось маргинальным хобби, и гарантия Debian была такова: если вы сломаете это, вам достанутся обе половины. Теперь большинство дистрибутивов поставляются с набором графических инструментов, преобразующих нажатия кнопок, флажков и манипуляции с полосами прокрутки в значения во все тех же .conf файлах. Но файлы остаются, вот почему современная система Linux пригодна и для новых пользователей, и хакеры ею довольны.

/mnt

Если вы вставите в компьютер CD, дискету или подключите переносной диск в USB-порт, то иконка носителя выскочит на рабочий стол или в меню Переход в Gnome так же шустро, как продавец сельской лавочки к покупателю. Теперь вы можете дважды щелкнуть на ней и получить доступ к содержимому при помощи вашего менеджера файлов. Если устройство не появилось там, где ожидалось, лучше всего поискать его в **/mnt** (от mount). Это каталог, в котором устройство подключается к иерархии Linux (учтите, некоторые дистрибутивы могут запутать вас, помещая сменные носители вроде CD и DVD в **/media**). Однако, поскольку мы живем в XXI веке, эти операции не должны требовать дополнительной помощи – ваши диски должны появляться магически.

Если вы все еще используете двойную загрузку Windows и Linux, то ваши Windows-диски также, скорее всего, появятся в каталоге **/mnt**. Однако вам следует знать, что хотя большинство дистрибутивов Linux без проблем осуществляют чтение/запись в старой файловой системе Windows FAT32, они не умеют работать с файловой системой NTFS, используемой большинством систем XP. [Это не совсем так: NTFS прекрасно поддерживается в режиме чтения. С записью данных на нее, действительно, могут возникнуть проблемы, – прим. ред.] Есть и обходные пути, основанные на свободных программах, что с каждым днем становятся лучше, но пока еще не слишком дружелюбны к начинающим.

/dev

Желающим убедиться, что все в системе является файлом, лучшее всего смотреть именно в этом каталоге. Здесь будет несколько файлов, соответствующих разделам вашего жесткого диска (**hda1**, **hda2**), дисководу (**fd0**), CD (**cdrom**) и любому другому устройству (**/dev** – от device [устройство]), подключенному к компьютеру. Но не думайте, что они статичны, только потому, что мы говорим о файлах. Например, все, что записывается в файл **/dsp**, будет переправлено через вашу звуковую систему на колонки. Чтобы увидеть этот эффект, вы можете взять любой файл, скормить его звуковой системе и услышать, как он звучит. Наберите

```
cat /etc/fstab > /dev/dsp
```



› **Дистрибутивы на CD, вроде Puppy Linux, имеют ту же структуру каталогов, только без каталога /home..**

и услышите некий шум из динамиков (сначала убедитесь, что они не на максимуме!): это речи – интересные и бессмысленные одновременно – текстового файла!

/proc

Это странная псевдофайловая система, находящаяся в памяти ОС и потому никогда реально не сохраняемая на физическом устройстве. proc происходит от process, потому что она хранит сведения обо всех процессах, запущенных на вашем компьютере. Процессами может быть все что угодно: от открытого web-браузера до работающего демона NTP, синхронизирующего часы вашего ПК с сервером в сети. То есть это динамический каталог, новая версия которого создается при каждой загрузке [на самом деле, содержимое каждого файла и каталога в **/proc** определяется непосредственно в момент их чтения и определяет самую свежую на тот момент информацию о состоянии вашей системы, – прим. ред.].

/var

Этот каталог является пристанищем данных, отправляемых из приложений на принтер. Интересного тут мало [спул принтера – лишь малая часть обязанностей каталога **/var**. В нем также хранятся системные журналы (**/var/log**), сведения о запущенных демонах (**/var/run**) и принятые сервером почтовые сообщения (**/var/mail** или **/var/spool/mail**), а в некоторых системах – документы, доступные через web-сервер, – прим. ред.].

/boot

После **/**, это первый каталог, к которому обращается компьютер после старта. Кроме файлов, связанных с вашим менеджером загрузки (*Lilo* или *Grub*), Linux хранит здесь действующее ядро ОС. Оставьте его в покое.

/lib

Любой, кто работал с Windows хоть какое-то время, сталкивался с DLL. DLL – это сокращение от dynamic linked library [динамически подключаемая библиотека]. Подобные библиотеки имеются и в Linux, отсюда и **/lib**. Библиотеки – часть программного обеспечения, необходимая для работы других программ. Например, *Gimp* может открывать и записывать JPEG файлы благодаря библиотеке *jpg*, содержащей необходимую информацию для работы с этим форматом. Более того, другие приложения могут также использовать данную библиотеку. Навряд ли вы загляните в этот каталог, разве что из любопытства.

/lost+found

Этот каталог, который, кажется, назвал Баз Лурхман [Baz Lurhmann], вы должны посетить в первую очередь в случае перезагрузки после сбоя или нештатного окончания работы (например, из-за отключения питания). Linux любит выключаться правильно, а именно: все файловые системы размонтируются и все процессы уничтожаются до того, как штепсель выдернут из розетки (это одна из причин, почему можно не дефрагментировать ваш диск каждую неделю) [здесь автор заблуждается, – прим.ред.]. Таким образом, когда случается неожиданное выключение, Linux требует восстановления системы. После катастрофического сбоя первым произойдет запуск системы fsck, программы, которая просматривает ваши диски в поисках поврежденных файлов. Она попытается исправить любые найденные повреждения и поместит их в **/lost+found**.

/tmp

В Windows имеется каталог **Temp**, в Linux – **/tmp**. В этом месте ОС хранит данные, необходимые ей временно, типа файлов, загружаемых из Интернета и открытых в приложениях, но еще не сохраненные в обычном месте. Большинство Linux систем очищает этот каталог – удаляя все содержимое – во время загрузки, так что ничего в нем не храните!

Большинство из описанных в этом кратком обзоре каталогов мало интересны тем пользователям, которые хотят просто использовать свой компьютер, и верно также то, что вам не нужно понимать структуру ОС, чтобы продуктивно работать. Однако даже краткий экскурс под капот вашей системы – очень полезное упражнение. Поначалу он должен подвигнуть вас на дальнейшее изучение и эксперименты с вашим компьютером, чего отнюдь не советуют делать пользователям Windows и OS X. Не исключено, что вы доэкспериментируетесь до поломки компьютера [не физической – ваше оборудование останется в целости, но систему придется чинить или переустанавливать, – прим. ред.], что может быть весьма обидно. Вот прекрасный повод изучить одну из возможностей виртуализации, имеющихся в Linux. С каким-нибудь бесплатным *VMware Player* (**www.vmware.com/products/player**) вы можете играть и ломать все, что хотите, а потом отключите виртуальное устройство и перезагрузитесь по новой. Чем не отличная игра! LXF

Что за штука... журналируемая файловая система?

Описанная в данном уроке файловая система – это структура файлов Linux, т.е. схема того, как они расположены в операционной системе. Можете еще почитать о форматах файловых систем типа ext3 и ReiserFS (некорректно называемых просто 'файловые системы'), являющихся структурами, разработанными для хранения и чтения данных с устройств вроде дисков и флэш-драйвов. Важно, чтобы данные, записанные на устройство, совпадали с данными, считанными с устройства – чтобы пользователь мог довериться целостности файловой системы. Файловая система также следит сама за собой: сколько осталось свободного места и так далее.

Журналируемая файловая система более пристально следит за деталями (называемыми метаданными), и при записи части информации выполняется несколько действий: данным выделяются блоки, указатель обновляется, обновляется размер файла, а затем записываются данные. Если этот процесс прервать, то нежурналируемая файловая система может оказаться «противоречивой», и *fsck* (см. **/lost+found** выше) при перезагрузке попытается увязать эти противоречия вместе. Журналируемая файловая система использует метод наподобие базы данных для ведения журнала метаданных (это все, что выполняется с файлами, кроме их содержимого) как «атомарных транзакций». После сбоя этот журнал может быть воспроизведен заново, чтобы реконструировать данные намного быстрее и с большей надежностью, чем у нежурналируемой системы.

В отличие от Windows, журналируемая файловая система постоянно дефрагментируется, так что проблемы, связанные с фрагментацией данных (когда один файл разбросан по множеству областей на всем диске), не возникают [самодефрагментация никак не связана с журналированием; просто большая часть современных файловых систем для Linux стремятся делать и то, и другое, – прим. ред.]. Проблемы фрагментации могут привести к замедлению работы системы, неэффективному использованию дискового пространства и потере файлов. Имеется порядочно журналируемых файловых систем, доступных в дистрибутивах Linux, но чаще всего встречаются ext3 или ReiserFS.

» **Через месяц** Сделайте первые шаги в настройке простого файл-сервера.

Hamachi: установка



Ghamachi – это графическая оболочка приложения *Hamachi* для среды Gnome. *Hamachi* – приложение с открытым исходным кодом для установки и настройки виртуальной частной сети. *Hamachi* прост в установке, использовании, легко настраивается… и не имеет ничего общего с суши. **Рэйчел Проберт** покажет нам, как его использовать.

Если вы разбираетесь в компьютерах лучше своих соседей, они считают вас экспертом. Каждому из нас иногда приходится побыть в этой роли, помогая друзьям и семье. И хотя я не отказываю никому в помощи, иногда совершенно не хочется выходить на улицу – особенно когда ты сидишь дома в теплой пижаме. Наш друг *Hamachi*, открытое приложение для настройки VPN, поможет нам остаться дома, когда кто-то просит помочь с настройкой своего компьютера. Почему бы не использовать для этого VNC? [Virtual Network Computing (VNC) – система удаленного доступа к рабочему столу компьютера, использующая протокол RFB (Remote FrameBuffer), – прим.

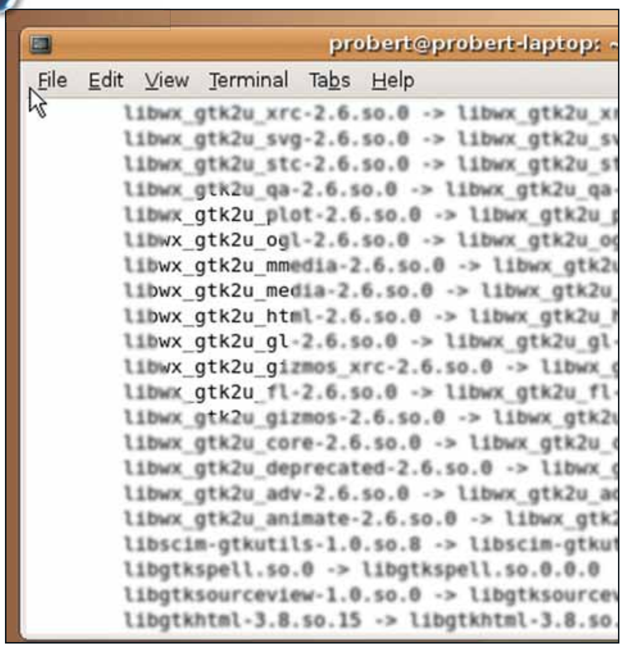


Наш эксперт

Рейчел Проберт (Rachel Probert)

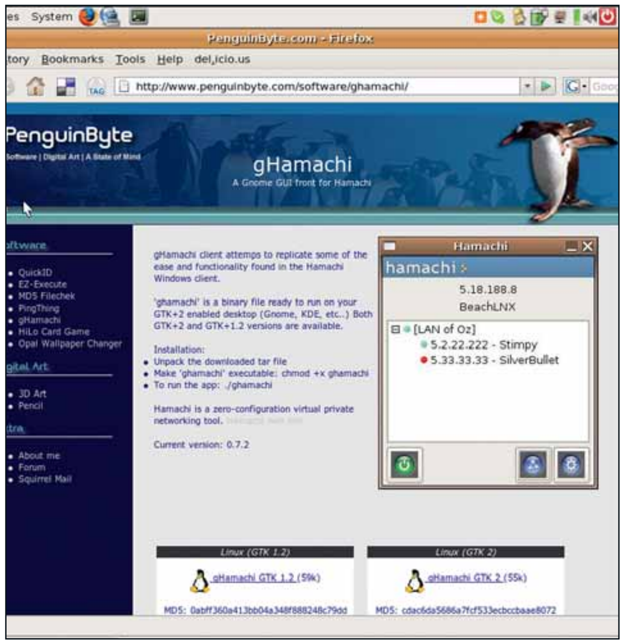
Последние семь лет консультирует развивающиеся фирмы и создает для них web-приложения.

Шаг за шагом: Установка и запуск GHamachi



1 Проверяем версию GTK

Ghamachi использует набор виджетов *GTK* версии 1.2 или 2. Как вы, наверное, знаете, *GTK* (или *Gimp Toolkit*) – это набор библиотек для создания приложений с графическим интерфейсом. Если вы не знаете, какая версия *GTK* используется в вашей системе, выполните команду ldconfig -v | grep gtk. Она выведет список файлов *GTK*, имена которых будут завершаться единицей или двойкой.



2 Загружаем GHamachi

Загрузите *Ghamachi* с www.penguinbyte.com/software/ghamachi, (в соответствии с вашей версией GTK) и сохраните на диск. Распакуйте архив на рабочий стол – там должна появиться иконка *Ghamachi*. Откройте свойства файла, выбрав пункт Свойства (Properties) в контекстном меню. Сделайте его исполняемым, выбрав Исполняемый (Execute) на вкладке Права доступа (Permissions).

защищенной VPN

перев.] Дело в том, что, используя VNC, вы открыты для атак извне. Однако с *Hamachi* и VNC все проблемы будут решены.

В этой статье мы рассмотрим процесс установки *Ghamachi*, графической оболочки *Hamachi* для среды Gnome. Это приложение не имеет всех возможностей консольной версии, но поможет настроить VPN тем пользователям, которые не очень любят работать в командной строке. Для того чтобы настроить безопасный доступ к удаленному рабочему столу, мы будем использовать *Ghamachi* совместно с VNC. Мы загрузим, установим и настроим эти приложения.

Что же такое *Hamachi*? Любители суши узнают в *Hamachi* рыбку с желтым хвостиком наподобие тунца, а мы – очень удобное приложение для настройки виртуальной частной сети (VPN). Это означает, что мы можем объединить несколько удаленных друг от друга компьютеров в защищенную сеть, как если бы они были соединены обычным сетевым кабелем в локальной сети. То, что делает *Hamachi* таким замечательным – это очень простая установка, настройка и использование. Программу может использовать кто угодно – объяснить, что к чему, можно даже по телефону.

Сразу после его установки и запуска вы сможете подключиться к другому компьютеру через VNC и немедленно оказать помощь. Вам даже не придется открывать порты на брандмауэре, потому что сервер *Hamachi* функционирует как «служба приглашений». Так, оба участника диалога открывают исходящее соединение с сервером *Hamachi*, который представляет их друг другу, используя открытые ими «эфемерные» порты. После установки соединения между сторонами на сервере больше нет необходимости. Теперь, когда вы знаете, как это работает, перейдем к делу.

Команды Hamachi

Если хотите копнуть поглубже, можете попробовать команды для работы из командной строки. Их названия говорят сами за себя.

- *hamachi start*
- *hamachi stop*
- *hamachi set-nick <nickname>*
- *hamachi login*
- *hamachi logout*
- *hamachi list*
- *hamachi get-nicks*
- *hamachi create <networkname> <password>*
- *hamachi delete <networkname>*
- *hamachi evict <networkname> <5.x.x.x>*
- *hamachi join <networkname> <password>*
- *hamachi leave <networkname>*
- *hamachi go-online <networkname>*
- *hamachi go-offline <networkname>*

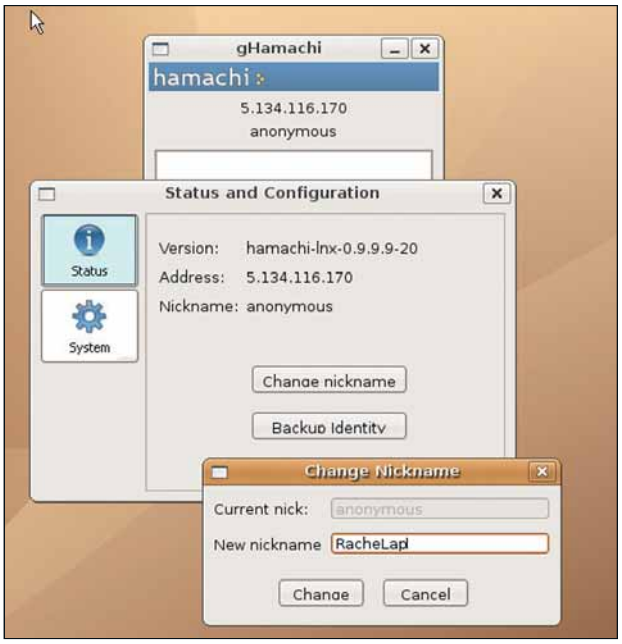
LINUX FORMAT На DVD

- Hamachi 0.9.9.9
- GHamachi 0.7.2



3 Устанавливаем сетевые драйверы

Когда вы попробуете запустить *Ghamachi*, то увидите окно с сообщением о том, что *Hamachi* требуется драйвер устройства TAP/TUN. Это означает, что для работы приложения нам потребуется установить драйвер виртуальной частной сети. Нажмите Да на запрос системы о загрузке '*tuncfg*' в режиме *sudo* и введите свой пароль *sudo* для продолжения установки.



4 Определяем сетевое имя

После запуска *Ghamachi* вы увидите окошко с IP-адресом, который выделило вам приложение. *Hamachi* использует адреса из семейства 5.0.0.0 для первого входа в систему. Так как доступ по IP-адресу не очень удобен для других участников системы, нужно задать сетевое имя для нашего компьютера. Для этого в окне настроек программы (кнопка Настроить Hamachi – Configure Hamachi) нажмите кнопку Изменить сетевое имя (Change Nickname).

Скорая помощь

Пользователи KDE могут использовать *Quamachi* – версию *Hamachi* для этой графической среды. Загрузить ее можно с сайта http://quamachi.sourceforge.net.

»



5 Создаем сеть

Нам необходима сеть, в которую могли бы войти другие пользователи, например, друзья, которым нужна ваша техническая поддержка. Для этого нажмите кнопку Создать сеть / Присоединиться к сети (Create Or Join Networks) и выберите вариант Создать сеть (Create Network). Введите имя сети. Это имя будут использовать другие пользователи для подключения к ней. Чтобы избежать нежелательных гостей, установите пароль. Советы по созданию паролей приведены в статье «15 ценных советов по безопасности» в LXF91.

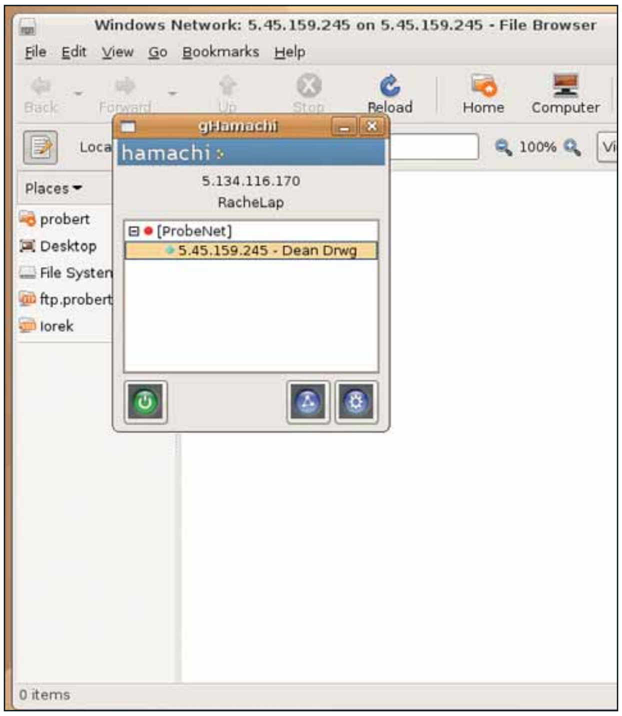


6 Настраиваем другие компьютеры

Сейчас нам нужно задать те же самые настройки на другом компьютере. К счастью, *Hamachi* прекрасно работает на различных платформах: Linux, Windows и Mac OS X, и вы можете использовать ее для связи компьютеров, на которых установлены разные операционные системы. Загрузить подходящую версию приложения можно на странице **www.hamachi.cc/download/list.php**. После установки приложения запустите его и задайте сетевое имя точно так же, как мы делали это на шаге 4.

Скорая помощь

Если вы хотите поподробнее узнать о том, как работает *Hamachi*, подпишитесь на подкаст Стива Гибсона 'Hamachi Rocks' на страничке **www.grc.com/SecurityNow.htm.**



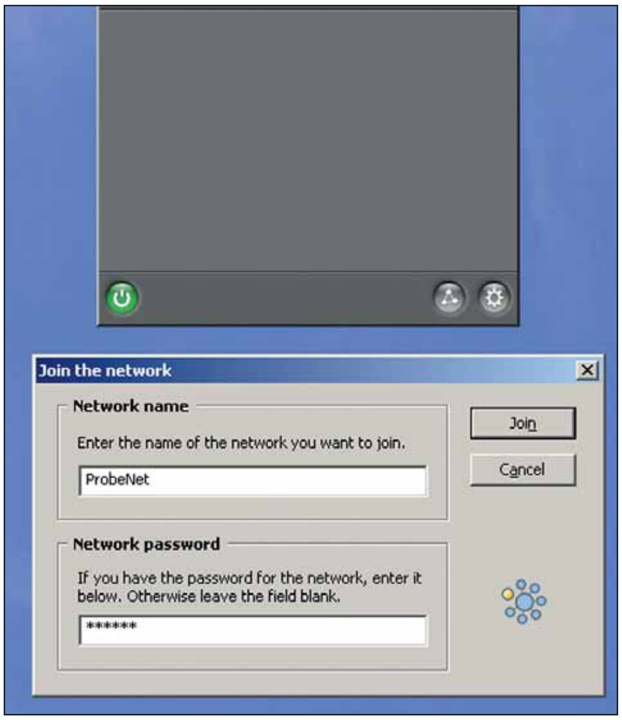
9 Обозреваем сеть

Сейчас вы должны увидеть других участников вашей сети. Это небольшое упражнение позволит вам не покидать своего дома, когда кто-нибудь опять призовет на помощь техподдержку. Щелкните правой кнопкой на имени пользователя и выберите пункт меню Обзор (Browse), чтобы задать общие каталоги для обмена музыкой или воспользоваться дорогим лазерным принтером вашего друга!



10 Отключаем пользователей

Если вы владелец сети, вы можете отключить пользователя от вашей сети. Для этого щелкните правой кнопкой на имени этого пользователя и выберите пункт меню Отключить пользователя (Evict User). Пользователь будет удален из сети. Конечно, он сможет подключиться снова, так как знает пароль, поэтому не забудьте изменить пароль, если вы твердо решили указать ему на дверь.



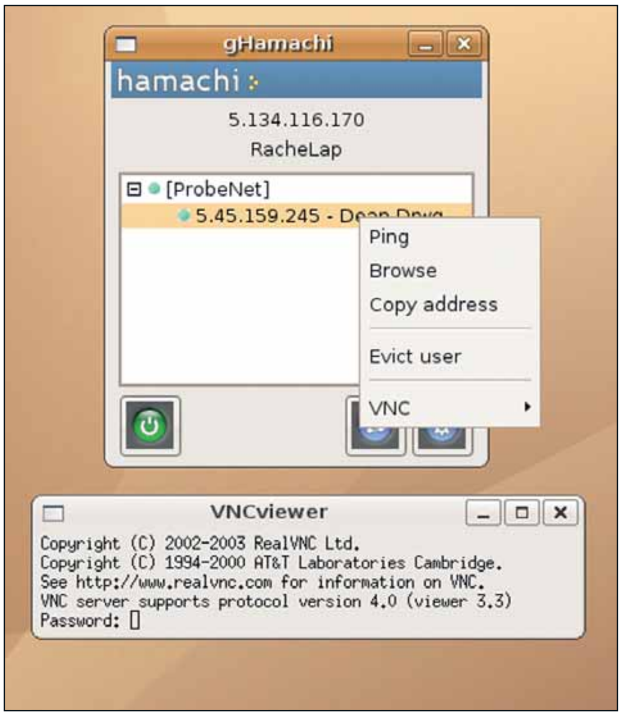
7 Присоединяемся к сети

Теперь мой друг Dean Drwg может присоединиться к моей сети. На втором компьютере нажмите кнопку Создать сеть / Присоединиться к сети (Create Or Join Networks) и выберите вариант Присоединиться к существующей сети (Join Existing Network). Введите имя сети и пароль, которые мы задавали на шаге 5, и нажмите на кнопку Присоединиться (Join). На первом компьютере в списке адресов сети появятся IP-адрес нового пользователя, начинающийся с 5, и имя пользователя. Для обновления этой информации может потребоваться перезапустить Hamachi.



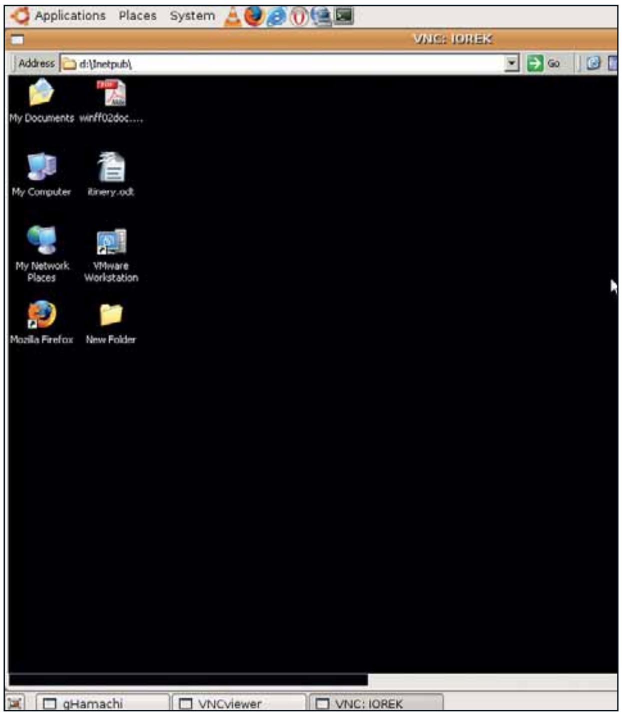
8 Если что-то идет не так

Если появились какие-то проблемы, щелкните правой кнопкой на названии сети и выберите пункт Установить соединение (Go Online). Если установить соединение не получается, и вы уже пробовали перезапускать *Hamachi*, щелкните правой кнопкой на IP-адресе одного из пользователей и выберите пункт Пинг (Ping). Не забудьте нажать Ctrl+C для останова. Если в окошке с пингом отображается сообщение «хост недоступен» ('host unreachable error'), то вероятна проблема с виртуальной сетевой картой (virtual network card). Ее решение можно поискать на форумах http://forums.hamachi.cc и www.ubuntuforums.org.



11 Подключаемся к другому компьютеру

Поскольку VNC – кроссплатформенная технология, ее можно использовать совместно с *Hamachi* для подключения к другому компьютеру независимо от установленной на нем ОС. Если она у вас уже установлена, вы можете выбрать пункт меню VNC > Подключиться к этому компьютеру (VNC > Control This PC). Вам потребуется ввести пароль второго компьютера в командной строке.



12 Дело сделано

Вот и все. Вам больше не придется покидать свою берлогу, чтобы помочь друзьям или родственникам. Теперь у вас есть безопасный сетевой доступ к другим компьютерам. *Hamachi* использует защищенное соединение и методы аутентификации, основанные на промышленных стандартах и протоколах. Никто не получит доступа к соединению между двумя компьютерами, установленному с помощью *Hamachi*. LXF

Скорая помощь

На досуге почитайте HamachiWiki – www.hamachiwiki.com. Вы найдете море полезных статей по использованию Hamachi в самых разных целях, начиная от сетевых игр и общего доступа к файлам и заканчивая совместным ведением бизнеса.

Web: Сделай свой

Вам не надо быть гуру, чтобы сделать его самому. **Майк Сондерс** даст вам инструменты и навыки, необходимые для создания небольшого web-приложения.





Наш эксперт

Майк Сондерс поддерживает сайт *Linux Format* и программирует со времен ZX Spectrum. В данный момент цель его жизни – изучить все известные человечеству варианты ассемблера.

Если я скажу «разработка программного обеспечения», вам на ум, возможно, придут часы, проведенные в текстовых редакторах и командной строке за магическими трюками, на постижение которых ушли годы. И зря! Чтобы создать программу, вам не нужны ни 600-страничная книга, ни знание бесчисленных флагов GCC – вот это я и попытаюсь продемонстрировать на данном уроке. Мы сделаем нечто реально клевое: каталогизатор музыкальных дисков, работающий через браузер, который вы по мере обучения сможете расширять. К концу урока вы сумеете приспособить его для учета DVD, игр и даже, если угодно, хомячков.

Но как же мы создадим программу, не программируя? Ответ прост: заставим чужие программы выполнить всю черновую работу за нас. Воспользуемся *Nvu* для проектирования интерфейса нашего приложения, а также OpenOffice.org Base как хранилищем данных о нашей CD-коллекции. Мы свяжем их с помощью небольшого скрипта на PHP, имеющегося на нашем диске – это очень просто. Даже если вы сроду не написали на PHP ни строчки, вы все равно поймете, о чем речь, из комментариев в коде.

Наше руководство создаст вам твердую почву для разработки сетевых приложений, работающих с базой данных, и создания форм с помощью HTML-редактора. Конечный проект будет работать на вашем PC, доступный через браузер. Познакомившись с тем, как он делает это, вы сможете выложить приложение в Сеть на суд своих друзей. А всего лучше, что мы не будем скучать, а создадим что-то с нуля и изучим различные инструменты. Хватит болтать – начнем!

> «Познакомившись с тем, как он работает, можете выложить его в сеть на суд друзей.»

Экипируемся

Прежде всего нужно установить программы, которые мы будем использовать для создания каталогизатора дисков:

» *Apache*. Самый популярный web-сервер. Наша программа будет работать внутри браузера, и ее страничку надо обслуживать; а Apache это здорово умеет.

» *MySQL*. СУБД для хранения данных о музыкальных дисках. Мы не будем напрямую взаимодействовать с MySQL, пусть с ней управляется другая программа...

» *OpenOffice.org*. Компонент *Base* из этого пакета позволит нам создавать и управлять базами данных, не прикасаясь к командной строке.

» *Nvu*. Web-редактор на движке Mozilla. *Nvu* позволит создать главное окно программы, не задумываясь об HTML-тэгах.

» *PHP*. Популярнейший язык программирования сценариев, используемый многими сайтами в сети. Мы будем использовать его для связи интерфейса страницы с содержимым базы данных.

Эти программы должны быть доступны в вашем менеджере пакетов (Synaptic в Ubuntu, Control Center в Mandriva, Pirut в Fedora, Yast в SUSE) за несколько щелчков мыши. Если у вас возникли проблемы с их поиском или нет выхода в Интернет, можете взять их с нашего DVD: *Apache* и *MySQL* находятся в разделе **Сервер**, *OpenOffice.org* в разделе Офис, PHP в разделе **Разработка**, а *Nvu* в разделе **Интернет**. Однако, простоты ради, рекомендую вам по возможности использовать менеджер пакетов – это избавляет от ручной компиляции и установки.

Чтобы получить и установить требуемые пакеты через Интернет в Ubuntu, введите в терминале следующую команду:

```
sudo apt-get install apache2 libapache2-mod-php5 mysql-server
openoffice.org nvu
```

В других дистрибутивах поищите в вашем менеджере пакетов *Apache* и *MySQL* (никаких библиотек разработчика не потребуется). Обычно они называются *mysql*, *mysql-server*, *apache*, *apache2* и т.п. Во

Другие HTML-редакторы

Я использовал *Nvu*, потому что с ним легко работать, и он основан на *Mozilla*. Если вдруг вы от души ненавидите *Nvu*, можете испытать другие HTML-редакторы, при условии, что будете вставлять скрипт **control.php** до тэга </body> в конце. Наиболее критичные участки HTML-кода – имена полей формы. Как отмечается на уроке, они должны совпадать с именами, используемыми в базе данных. Например, текстовое поле ввода названия CD в HTML должно называться CDName:

```
Name: <input name="CDName"><br />
```

Используете ли вы другой визуальный редактор или же HTML-редактор вроде *Quanta Plus*, очень важно давать полям те имена, которые указаны в базе. В противном случае PHP-скрипт не будет знать, куда вставлять данные; он берет имена полей на нашей web-страничке и вносит по ним информацию в базу данных. Придерживайтесь имен, указанных здесь, и у вас не будет никаких проблем.

CD-менеджер

всех дистрибутивах *Apache* и *MySQL* после установки должны стартовать автоматически. Но для уверенности откройте терминал и переключитесь в режим пользователя root (*sudo bash* в Ubuntu и *su* в других дистрибутивах), затем наберите

```
/etc/init.d/apache2 start
/etc/init.d/mysql start
```

Если получите сообщение о не найденном файле, замените в первой строке apache2 на apache: имя скрипта для запуска *Apache* может отличаться в разных дистрибутивах.

Создаем интерфейс

Итак, мы установили вспомогательные серверные программы и запустили их. Теперь спроектируем интерфейс приложения! С его помощью вы сможете просматривать содержимое базы данных и добавлять и удалять информацию о дисках через текстовые поля. Запустите *Nvu*, и увидите пустую страницу, готовую для редактирования – если нет, выберите в меню File > New. В правой панели *Nvu* введите что-нибудь наподобие Каталогизатор CD 1.0, выделите набранный текст и нажмите на панели инструментов кнопку B, чтобы шрифт стал жирным. Вот вам и заголовок нашей программы.

Далее необходимо создать форму для ввода информации о дисках, добавляемых в базу . Нажмите Form на самой верхней панели инструментов, введите Main в качестве имени и manager.php в качестве Action URL, затем выберите метод POST . Наконец, нажмите OK, и увидите подсвеченную синим область на главной странице. Теперь можно добавлять к интерфейсу текстовые поля – именно их мы используем в готовой программе для внесения в базу информации о новых дисках.

Напечатайте Name, затем выберите Insert > Form > Form Field. Укажите Text в качестве типа поля, затем введите CDName в поле Filed Name. Заметим, что строка CDName очень важна: имена полей текстовых элементов ввода должны соответствовать заголовкам, используемым в базе данных, которую мы скоро создадим. Значение Initial Box устанавливать не надо: просто нажмите OK, и увидите текстовый компонент на странице.

Итак, у нас есть куда вводить информацию об альбоме в законченном продукте. Теперь не мешает добавить еще одно текстовое поле, на сей раз для его автора. Повторите шаги предыдущего абзаца, но на этот раз перед вставкой поля формы введите Author: и назовите его CDAuthor, а не CDName. Теперь у нас есть два текстовых поля ввода на главной странице. Повторите инструкции предыдущего параграфа, но теперь введите перед полем CDDate:, а само поле назовите NewDate. Все понятно? Итого, у вас должны быть три текстовых поля с именами CDName, CDAuthor, CDDate с метками Name:, Author:, Date: перед ними.

Давайте добавим поле описания: мало ли что еще можно сказать про диск! После поля Date: введите Description:, затем выберите в меню Insert > Form > Text Area. Назовите новое поле CDDesc в Field Name и установите размер в 5 строк на 40 столбцов, а затем нажмите OK. Наконец, нам нужна кнопка отправки, чтобы занести информацию о диске в базу после ее ввода. Перейдите в Insert > Form > Form Field, затем выберите Submit Button из ниспадающего списка Field Type и введите Add Entry в окошке Field Value. По завершении нажмите OK.

Теперь ваша форма должна выглядеть как на **рис. 1**. Поздравляем – вы создали интерфейс! Конечно, сейчас он простоват, но вы можете украсить его цветом, картинками и чем пожелаете, применив средства *Nvu*. Выполните команду File > Save As, введите CD Manager в качестве заголовка страницы (Page Title), а затем сохраните ее как **manager.php** (не **manager.html**, так как позже мы вставим туда PHP-код).

Если, по-вашему, что-то пошло не так, проверьте файл **manager.php** в разделе **Журнал/WebDev** на нашем диске; он содержит HTML-код, который вы пытались создать. Откройте его в текстовом редакторе и сверьте с тем, что у вас получилось, чтобы ничего не упустить.



Время делать базу

Разработав «лицо» программы, займемся внутренней частью. Теперь наша задача – использовать *MySQL* для создания базы данных, в которой мы будем хранить информацию о CD. Возможно, вы удивляетесь, почему в таком маленьком проекте мы не пишем данные просто в текстовый файл. Но ведь потребуется писать процедуру обработки текстовых файлов! Куда проще установить *MySQL*, накидать туда данных, и пусть выполняет всю остальную работу. Откройте терминал и переключитесь в режим root (*sudo bash* или просто *su*). Затем введите

```
mysql -uroot -p
```

Вы получите права администратора на доступ к базам *MySQL*. Большинство дистрибутивов по умолчанию не устанавливают пароль для *MySQL*, и в ответ на запрос можете просто нажать Enter. Если вы установили пароль до этого – введите его. В командной строке *MySQL* наберите

```
create database manager;
```

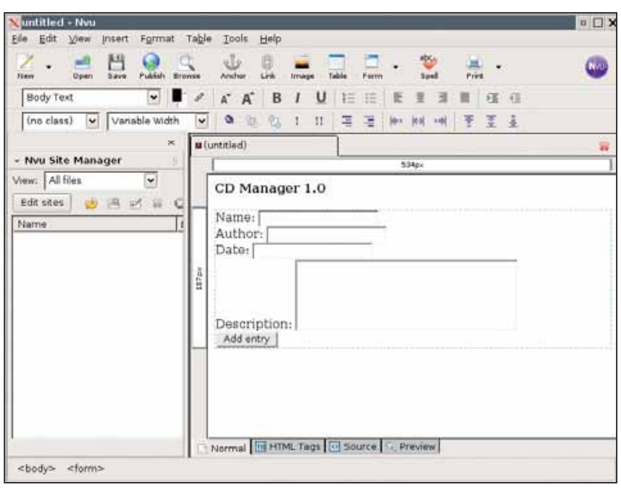
Эта команда создаст новую базу данных для нашего каталога.

Нажмите Ctrl+D, чтобы выйти из командной строки *MySQL*. Теперь нужно установить специальный компонент для связи *Openoffice.org* и *MySQL*; скачиваем **mysql-connector-java-5.0.5.tar.gz** с **www.mysql.com/products/connector/j** и сохраняем его в домашний каталог. Версию 3.1 не берите, она слишком стара.

Вернитесь в командную строку (проверьте, что вы все еще root) и введите

```
tar xfvz mysql-connector-java-5.0.5.tar.gz -C /opt/
```

Эта команда извлечет инструмент «коннектор» в каталог **/opt** ваше- »



› **Рис. 1. Так должны выглядеть ваши текстовые поля и кнопка посылки данных Add Entry.**

» го компьютера. Теперь запустите *OpenOffice.org* и перейдите в Tools > Options. Выберите пункт Java в подменю *OpenOffice.org* слева, затем на кнопку Class Path справа, затем Add Archive. Перейдите в каталог **/opt/mysql-connector-java-5.0.5**, укажите файл **mysql-connector-java-5.0.5-bin.jar** и щелкните кнопку Open. Нажмите OK и снова OK, чтобы убрать два появившихся диалога, и ваш коннектор установлен. Наконец, перезапустите *OpenOffice.org*.

Откройте File > New > Database, и когда появится Мастер, выберите Connect To An Existing Database и укажите MySQL в выпадающем меню. Нажмите Next, и на очередном экране выберите соединение с помощью JDBC. Щелкните Next. Теперь будем заполнять базу информацией. Введите manager в качестве имени базы и localhost в качестве URL, нажмите Next и введите root в качестве имени пользователя. Наконец, нажмите Next, чтобы перейти на последний экран, а там нажмите Finish. Вас могут попросить ввести новое имя файла – просто укажите старое имя, так как мы в дальнейшем не будем его использовать.

Скорая помощь

Перед внесением изменений в структуру базы можете сделать ее резервную копию командой mysqldump -uroot -p manager > backup.sql. Для восстановления данных из резервной копии введите mysql -uroot -p manager < backup.sql. backup.

Вдарим по таблицам

Теперь выберите Create Table In Design View в панели Tasks в главном окне *OOo Base*. Появится сетка, в которой можно проектировать базу данных. В столбце Field Name введите ID, щелкните правой кнопкой мыши на квадрате слева (с зеленой стрелкой) и выберите пункт Primary Key. Щелкните на тип TEXT [VARCHAR] в поле справа от ID (под заголовком Field Type) и выберите Integer [INT]. Благодаря этому у добавляемых CD в нашей базе будет целочисленный индекс.

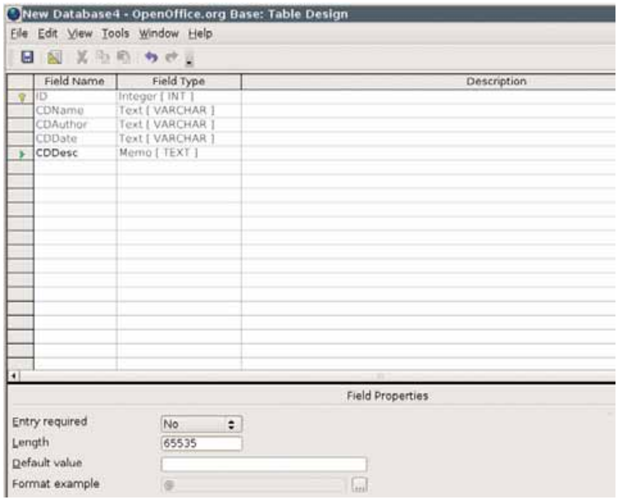
Далее надо сообщить имена текстовых полей, созданных с помощью *Nvu*, поэтому введите CDName, CDAuthor, CDDate, CDDesc вертикально под ID. Вы заметите, что OOo назначает этим полям базы тип данных Text [VARCHAR] под заголовком Field Type, то есть они могут содержать текст переменной длины.

Для поля CDDesc щелкните на поле Text [VARCHAR] и выберите Memo [TEXT] из выпадающего списка – тогда нам перепадет побольше пространства для хранения описания альбома. Теперь окно вашего *OOo Base* должно выглядеть как на **рис. 2**. Выберите в меню File > Save, введите entries в качестве имени таблицы и нажмите OK. Ура – мы сделали это! Теперь закроем *OOo* и перейдем к финальным стадиям.

Одну из настроек базы надо сменить, чтобы новые записи о дисках получали новые номера ID. Откройте терминал, переключитесь в режим root и введите mysql -uroot -p, чтобы войти в *MySQL* как было описано раньше. В строке *MySQL* введите

```
use manager;
alter table entries change id id int(10) auto_increment;
```

Первая команда активирует нашу базу manager, а вторая изменяет поле id в таблице – первую колонку, которую мы создали. Это строка



› Рис. 2. Убедитесь, что таблица базы данных в Base соответствует изображенной здесь!

снабжает номером каждую запись о CD, чтобы потом можно было легко их удалять. Сказать *MySQL* «удали запись номер 5» куда проще, чем «удали где-то там запись с текстом, содержащим слово Бах», особенно если вы поклонник Баха и у вас таких полно! К сожалению, *OpenOffice.org* не позволяет полю id увеличиваться каждый раз при добавлении новой записи, поэтому без описанного выше трюка он просто продолжал бы ставить 0 в поле id. Ключевое слово auto_increment говорит, что записям последовательно присваиваются номера 1, 2, 3 и так далее.

Давайте припомним свои достижения: мы спроектировали страницу, которая будет служить интерфейсом к каталогизатору дисков, включая форму для добавления новых альбомов. Затем мы использовали *MySQL* и *OpenOffice.org Base* для создания базы, соответствующей форме на нашей странице с тем, чтобы они могли передавать друг другу информацию. Короче, мы создали графическую оболочку нашей программы и внутреннюю часть для хранения данных, и все это ни написав ни строчки кода. Теперь добавим немного PHP, чтобы связать их вместе!

Войдите в раздел **Журнал/WebDev** на нашем диске и скопируйте файл **control.php** в свой домашний каталог. Теперь откройте в двух окнах редактора файлы **manager.php** и **control.php**, для переноса содержимого **control.php** в **manager.php**. Менять ничего не надо – просто скопируйте содержимое **control.php** и вставьте перед тэгом </body> в **manager.php**. Если вы забуксовали, просмотрите файл **manager-final.php** в разделе **Журнал/WebDev** на диске, это полная и окончательная

Разберемся с PHP

Файл **control.php**, который вы включили в вашу страницу **manager.php**, контролирует добавление и удаление записей о дисках. Объяснить PHP в этой статье вряд ли удастся, но все же стоит взглянуть на файл, чтобы понять, как можно расширить HTML и улучшить вашу программу. Поскольку код занимает всего 25 строк, это не так уж сложно.

Скрипт PHP подсоединяется к созданной базе, затем проверяет, была ли отправлена новая запись через механизм POST (как мы установили при создании формы). Он также смотрит, надо ли удалять запись. После этого сценарий получает все данные из базы и отображает их:

```
while ($r = mysql_fetch_assoc($result)) {
extract($r);
echo "<table border=\"1\" width=\"100%\"><tr>";
foreach ($r as $keys => $contents)
echo "<td><strong>$keys</strong>: $contents</td>";
echo "<td>(<a href=\"manager.php?delete=$id\">delete this
entry</a>)</td></tr></table><br />";
}
```

Желая улучшить внешний вид программы, вы можете начать с редактирования HTML в этом разделе. Здесь PHP-скрипт получает все записи из нашей базы, затем для каждой из них составляет таблицу <table> и отображает каждое поле записи в ячейке <td> (то есть CDName, CDAuthor и так далее). Для каждой записи скрипт предоставляет ссылку на удаление; нажмите ее, и скрипт перезагрузится, предварительно произведя удаление записи.

Если вы немного знакомы с HTML (руководства изобилуют в интернете), можете попробовать навести лоск на отображение данных. Учтите, что если вы будете редактировать файл в *Nvu*, то вам придется сохранить код PHP, а потом вставить его обратно, потому что *Nvu* постоянно его удаляет. Кроме того, придется переименовать файл в **manager.html**, чтобы *Nvu* смог его открыть.

И последнее: если вы редактируете HTML-тэги внутри кода PHP, и в тексте есть кавычки ("), надо экранировать их обратным слешем, чтобы PHP распознал их как текстовый символ для вывода, а не как часть кода скрипта. Вы можете увидеть, как это делается, в строке <a href... приведенного здесь кода.

версия того, что должно быть.

Попытка – не пытка!

Пора попробовать программу. Скопируйте переделанный файл **manager.php** (со вставленным кодом) в каталог документов *Apache*. Скорее всего, для этого потребуются права root. Местоположение документов web-сервера зависит от дистрибутива, но в большинстве случаев это **/var/www/html** – если вы не найдете этот каталог, заставьте ваш менеджер пакетов показать список файлов, установленных с *Apache*, и найдите каталог **www/html**. Например, чтобы скопировать обновленный файл **manager.php** из вашего домашнего каталога, переключитесь в режим root и введите

```
cp manager.php /var/www/html
```

Теперь, благословясь, откройте в браузере **http://localhost/manager.php** (localhost – это имя вашего компьютера). Если все хорошо, вы увидите созданную нами форму, готовую к использованию и обслуживаемую *Apache*. Попытайтесь добавить новый диск: заполните все поля, хотя бы случайным текстом, просто для тестирования, и нажмите на кнопке Add Entry. При перезагрузке страницы информация добавится в базу, и вы увидите ее в новой таблице внизу страницы (как на рис. 3). Йес! Заработало!

Вы также увидите, что PHP-скрипт создает ссылку delete для каждой записи. Она указывает на файл **manager.php**, но добавляет к URL запись delete=, позволяющую скрипту узнать, что надо удалить. Следуя врезке «*Разберемся с PHP*» (слева), если вы худо-бедно знакомы с HTML, можете открыть файл **manager.php** в текстовом редакторе и навести лоск на страницу.

Наша программа хоть и незатейлива, но хорошо работает с небольшой коллекцией CD. Но она способна на большее: можете расширить ее, включив списки дорожек или поле для URL-адреса обложки альбома. Возможно, музыкальных CD у вас немного, но зато огромная коллекция DVD, и вы хотите ее тоже каталогизировать. Так примените полученные знания для фильмов! Помните, что создавая новые поля ввода на форме или меняя имена существующих, вы также должны обновить соответствующие имена в базе данных.

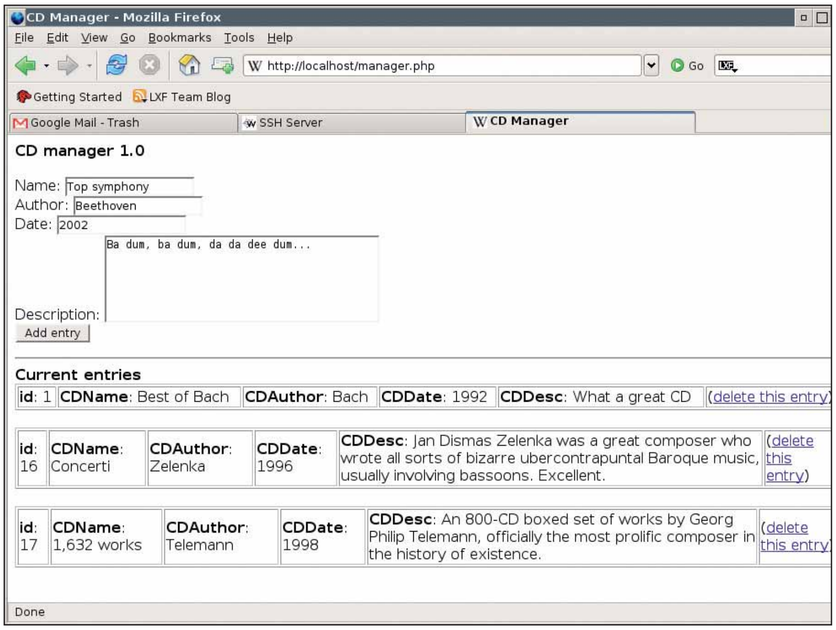
Если что-то пойдет не так и вы застрянете, не беда: полный код имеется на нашем DVD, и вы можете сравнить его со своей версией. Если и это не поможет, попытайте счастья на форумах **www.linuxforum.ru**. Подробно опишите вашу проблему, приложите код, с которым вы работаете, и мы посмотрим, чем можно помочь.

Поделитесь добром

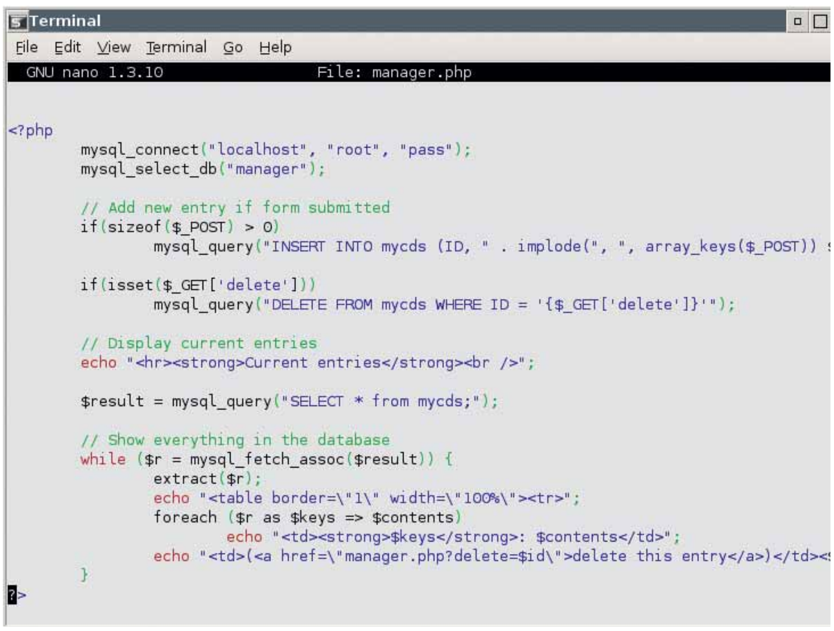
Так как наша программа работает через браузер, она не ограничивается только вашей личной машиной. Если у вас есть доступ к подходящему хосту, можете загрузить файл **manager.php** и воссоздать на нем базу, чтобы другие видеть вашу работу. Заметим, что сайт должен предоставлять поддержку PHP и *MySQL*: интернет-провайдеры обычно позволяют загружать только статическое содержимое, а вам нужно нечто посолиднее.

Инструменты настройки, используемые провайдерами, весьма разноообразны, и я не буду детально описывать процедуру загрузки. Просмотрите настройки и FAQ у вашего провайдера на предмет наличия упоминаний о *MySQL*. Вам необходим способ создать базу, как мы сделали с помощью *OpenOffice.org* – ваш хостер может предоставить для этого простые решения на основе браузера, а не то повозитесь с командной строкой *MySQL*. Есть тысячи руководств по ее использованию; Google вам в помощь!

Учтите, что PHP-скрипт спроектирован со спартанской лаконичностью, без всяких мер защиты (например, проверки, что введенные данные корректны и не содержат другие SQL-команды). Если вы поместите свой проект в сеть, сообщайте URL только людям, достойным доверия, в противном случае кто-то может попробовать взломать вашу базу. Простой способ отвадить людей от вставки данных – удалить строки с 5 по 11 в скрипте PHP (от if(sizeof) до mysql_query("DELETE FROM включительно). Когда захотите обновить или удалить записи, верните эти строки назад. Удачи в создании приложений! LXF



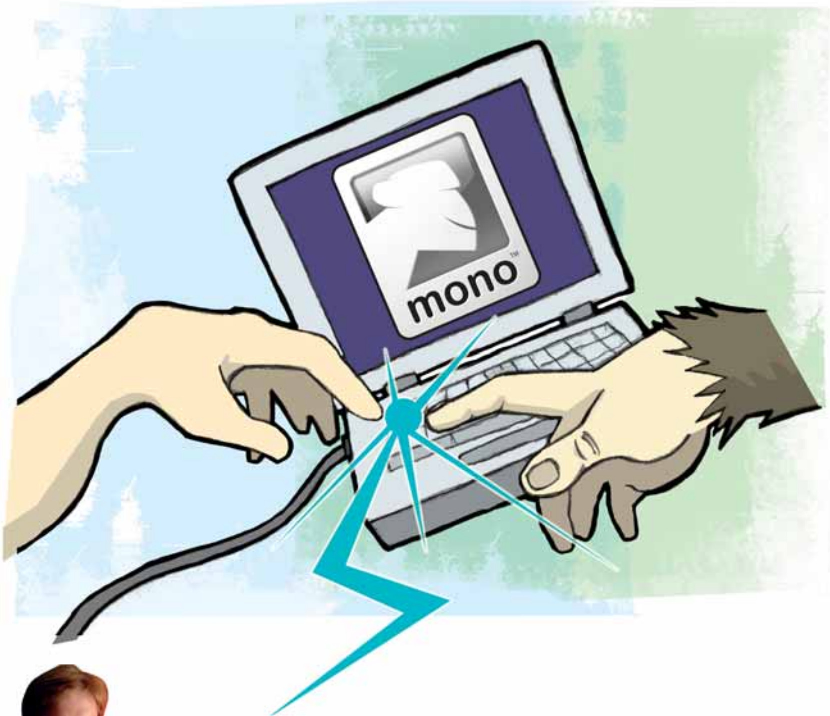
› Рис. 3. Наше приложение готово. Оно не особо шикарное, но вы можете добавить цветов, картинок и таблиц. 'Бо дун' – это, между прочим, музыкальный термин.



› Для редактирования файла manual.php возьмите редактор с подсветкой синтаксиса, типа Nano или Kate: проще будет читать код.

Mono: ООП для

Заявите свои права наследования в C#: на втором уроке по ООП **Пол Хадсон** покажет вам, как объекты передают свои знания из поколения в поколение.





Наш эксперт

Пол Хадсон полагает, что Mono – лучшая вещь со времен мультфильма Pinky and the Brain, и сейчас поддерживает два проекта на основе Mono на SourceForge

Что это была бы за жизнь, если бы мы были обречены Провидением следовать пути, проторенному предками? Тогда я готовил бы проповедь на воскресенье, а не писал эту статью! Но в жизни мы способны возвыситься над исходным состоянием, делая то, чего не дано нашим родителям. C# здесь имеет аналогию: один класс может «наследовать» от другого, то есть получать все методы и переменные своего родителя, но может и добавлять свои собственные. На самом деле, как один класс наследует от другого, так и тот класс может наследовать от другого класса, и так далее – вы можете продолжать цепочку до бесконечности!

Определим суть сегодняшнего учебника: в прошлый раз мы написали простую карточную игру, чтобы разобраться, как создавать собственные классы для решения реальных задач. Сегодня мы собираемся изготовить несложное подобие стратегической игры *Civilization*, концентрируясь на более мощных свойствах объектно-ориентированного программирования. Признаюсь, что игра – просто способ поддержать ваш интерес, чтобы вы по ходу дела подсознательно впитали много скучной теории; надеюсь, моя уловка сработает!

Прорисовываем экран

Проект у нас большой, и вариантов его начала предостаточно. Но лучше всего нырнуть прямо в игру, заставив ее загружаться и рисовать на экране, хотя «экран» покамест будет просто последовательностью пробелов: игра-то еще пуста! Поэтому запустите *MonoDevelop* и создайте новый консольный проект C# под названием *Snivilization*. Измените строки using наверху на следующие:

```
using System;
using System.Collections.Generic;
using System.Drawing;
```

Мы применим их позже. Вы также должны изменить название MainClass на Snivilization, так как это будет основной класс для нашей игры. Метод Main() в настоящий момент содержит строку Console.WriteLine(); удалите ее и вставьте следующие на ее место:

```
Snivilization game = new Snivilization();
game.Run();
```

Таким образом мы создадим класс Snivilization и вызовем пока не определенный метод Run(), по сути, запуская программу и передавая ей управление. Методу Run() необходимо войти в бесконечный цикл, чтобы игра продолжалась, пока пользователь не пожелает выйти. Вот черновой вариант метода Run():

```
void Run() {
  DrawScreen();
  while (true) {
    ConsoleKeyInfo key = Console.ReadKey(true);
    switch (key.Key) {
      case ConsoleKey.Escape:
         break;
      case ConsoleKey.C:
         DrawScreen();
         break;
    }
  }
}
```

DrawScreen() – это еще один пока не определенный метод, но мы скоро им займемся. Конструкция while(true) – бесконечный цикл в C#: он будет исполняться, пока условие истинно, а поскольку true и есть истина, вечная работа обеспечена. Остальная часть метода считывает одну



› *Civilization 4* Сида Майера: мы сделаем то же самое. Ну, не совсем точь-в-точь. Кому нужны идеально отрисованные галифе?

» **Месяц назад** Введение в объектно-ориентированное программирование.

создания игры

клавишу с консоли (передача true в качестве параметра ReadKey() означает «не позволять консоли делать ничего другого с клавишей»), затем проверяет, что это была за клавиша. Блок switch/case позволяет легко свериться с множеством значений, но пока нам нужно только вызвать метод DrawScreen(), если нажата клавиша C.

Метод DrawScreen() рисует много пробелов на экране, представляющем нашу карту. Вот он:

```
public void DrawScreen() {
   Console.Clear();
   for (int i = 0; i < GameHeight; ++i) {
      for (int j = 0; j < GameWidth; ++j) {
         Console.Write(" ");
      }
      Console.Write("\n");
   }
}
```

Переменные GameHeight и GameWidth новые, и надо добавить их описание – следующие две строки перед методом Main():

```
public int GameWidth = 100;
public int GameHeight = 40;
```

Перед запуском проекта необходимо сделать две вещи. Первое, щелкните правой кнопкой мыши на References и добавьте ссылку на System.Drawing. Второе, перейдите в меню Project > Options, выберите Runtime Options и проверьте, что выбрана среда выполнения .NET 2.0.

Теперь можно смело запускать программу, но – и это большое но! – не пытайтесь ее запустить с помощью F5. Панель вывода в *MonoDevelop* превосходна для вывода информации, но не является полноценным терминалом и поэтому не может считывать нажатия клавиш пользователем. Вместо F5 надо нажать F8, чтобы собрать проект, затем открыть терминал и перейти в каталог с проектом. Внутри подкаталога **bin/Debug** вы найдете **Snivilization.exe**, который можно запустить, набрав mono Snivilization.exe.

Сотворение мира

Вы-то видите пустой экран, но для жителей *Snivilization* мы только что создали небо и Землю, объявив «да будет свет»; а сейчас настало время создать из праха людей. Если вы раньше не играли в *Civilization*, знайте, что там есть два типа игровых объектов: города и боевые единицы. Оба они отличаются разнообразием: города могут быть выстроены в некоем стиле (Классический, Восточный и т.д.) и имеют различные размеры, а боевые единицы со временем эволюционируют, и вы можете сражаться чем угодно, начиная от Воина (первобытного мужика с дубиной) до межконтинентальных ракет. Но независимо от того, город ли это с десятимиллионным населением или танк, у всех этих объектов есть три общих свойства. Все они:

- имеют X- и Y-позиции на карте;
- должны обновляться каждый ход;
- должны прорисовываться на экране.

Общие свойства могут использоваться для создания базового класса, который будет предком обоих наших классов. В C# такие классы называются абстрактными, потому что им не соответствует конкретный объект: это просто определения. Так как все объекты нашей игры обладают перечисленными свойствами, создадим абстрактный класс SnivObject: пусть другие классы наследуют их от него. Вставьте следующий код до строки class Snivilization:

```
abstract class SnivObject {
   public int XPos;
   public int YPos;
   public abstract void Update();
   public abstract void Draw();
}
```

Объявление методов Update() и Draw() абстрактными означает, что любой класс – наследник SnivObject должен предоставить собственную реализацию Update() и Draw(), в противном случае Mono будет жаловаться. Фактически, это гарантия, что все объекты в игре могут прорисовывать себя сами. Мы можем проверить это, создав подкласс SnivObject с именем SnivCity. Для тех, кто не понял: подкласс наследует все методы и свойства базового класса, то есть класс SnivCity будет автоматически содержать переменные XPos и YPos.

Под определением SnivObject добавьте такой код:

```
class SnivCity : SnivObject {
   private float Size = 100;
   public override void Draw() {
      Console.Write("*");
   }
   public override void Update() { }
}
```

Обратите внимание, что класс определен как SnivCity : SnivObject – так C# сообщает, что «SnivCity наследует от SnivObject». Помните, что SnivCity должен иметь свои собственные (не абстрактные методы) Draw() и Update(). Пусты эти методы (как, например, метод Update()) или нет – неважно, лишь бы они существовали. Заметьте, что у SnivCity есть свойство Size, которого не было в SnivObject. Мы будем использовать его для отслеживания численности населения каждого объекта SnivCity.

Скорая помощь

Код в этом руководстве может сделать так, что у нескольких городов будут одинаковые координаты. Это предотвращается несложной магией на нашем диске. Обязательно загляните на диск!

Тайная инкапсуляция

Не каждый любит писать методы для всех своих переменных, тем более что это затрудняет понимание кода. Однако C# имеет опцию получше, известную как свойства. Они позволяют вам завести переменные, которые на самом деле являются функциями. Покажем, как это работает в практической – и довольно простой! – ситуации: добавьте переменную Name в класс SnivCity, чтобы мы могли идентифицировать города по имени.

Обычно это делается с помощью следующих строк:

```
public string Name;
```

Вместо этого мы собираемся включить открытую переменную Name и закрытую переменную _Name. Открытая переменная на самом деле будет свойством, то есть при обращении к ней она будет выполнять код. Вот как это выглядит в C#:

```
private string _Name;
public string Name {
   get {
      return _Name;
   }
   set {
      _Name = value;
   }
}
```

Таким образом, когда Name извлекается (то есть считывается), C# вернет значение, хранящееся в _Name. Когда Name записывается (при вызове set), C# сохранит значение в переменной Name: value – это специальное слово, означающее «значение value будет присвоено переменной Name». Конечно, этот пример не великое свершение, но легко представить, что в более сложных ситуациях вы можете захотеть вызвать другие функции при установке переменной.

Покажи и расскажи

Объекты, которые ничего не делают, бесполезны, поэтому с помощью нового класса SnivCity расширим Snivilization, чтобы игроки могли создавать города на карте. Нам при этом понадобится генератор случайных чисел (для размещения городов случайным образом), а также массив для хранения всех городов. Добавьте две строки кода после объявления переменных GameWidth и GameHeight:

```
public static Random Rand = new Random();
public List<SnivCity> Cities = new List<SnivCity>();
```

Тут нужен способ создавать города, но вместо реализации всего игрового движка мы просто позволим игре создать город по нажатию кнопки N. Примечание: если вы хотите расширить игру, вот вам хороший старт!

На данный момент блок switch/case проверяет только клавишу C, но несложно расширить его так, чтобы обрабатывалась клавиша N. Вот простой код:

```
case ConsoleKey.N:
    NewCity();
    DrawScreen();
    break;
```

Метод NewCity() новый, но ему всего лишь надо создать город, выдать ему произвольную позицию и добавить в список Cities:

```
public void NewCity() {
    SnivCity city = new SnivCity();
    city.XPos = Rand.Next(0, GameWidth);
    city.YPos = Rand.Next(0, GameHeight);
    Cities.Add(city);
}
```

Само по себе добавление города в массив фактически ничего не делает, так как город еще не видим на экране. Чтобы сделать наши города видимыми, надо изменить метод DrawScreen(): пусть проверяет, есть ли город в каждом поле, и если есть, сообщает городу, чтобы он себя прорисовал. Вы думаете, это потребует много кода? Отнюдь:

```
for (int i = 0; i < GameHeight; ++i) {
    for (int j = 0; j < GameWidth; ++j) {
        foundcity = null;
        foreach(SnivCity city in Cities) {
            if (city.YPos == i && city.XPos == j) {
                foundcity = city;
                break;
            }
        }
        if (foundcity != null) {
            foundcity.Draw();
        } else {
            Console.Write(" ");
        }
    }
    Console.Write("\n");
}
```

На данный момент в прорисовке городов интересного мало: были на поле пробелы, стали звездочки. Но будь у вас SDL, ваши города могли бы обзавестись графикой – рисовали бы свое имя, размер и так далее; поэтому подарить городам самостоятельную прорисовку –хорошая идея.

Попытайтесь-ка снова запустить программу – прогресс налицо: все по-прежнему начинается с пустого экрана, но стоит нажать N, как возникает новый город, готовый населить мир. Чем дальше, тем круче!

Проблемы возникновения Sniv-городов решены, но дальнейших перемен не просматривается: каждый город навеки обречен быть крошечной звездочкой на карте. Такой застой раздосадует даже самых неприхотливых игроков. Немного оживим процесс: сделаем так, чтобы города с каждым ходом росли при нажатии клавиши C, затем изменим метод Draw() для городов, поставив цвет отображения в зависимость от размера.

У наших городов уже есть метод Update(), и чтобы обновить любой город, достаточно вызывать для него этот метод. Итак, изменим действие клавиши C в блоке switch/case метода Run() следующим образом:

```
case ConsoleKey.C:
    UpdateCities();
    DrawScreen();
```

Те, кто читает эту серию уроков с первого выпуска, должно быть, уже представляют метод UpdateCities(). А если нет – вот он!

```
public void UpdateCities() {
    foreach (SnivCity city in Cities) {
        city.Update();
    }
}
```

Теперь наш класс SnivCity содержит метод Update(); и что же этот метод делает? Да ничего! Вы можете при обновлении заставить город делать все что угодно (в зависимости от степени реалистичности вашей игры); ну, а мы будем просто увеличивать его население:

```
public override void Update() {
    float growthrate = 25;
    Size *= 1 + (growthrate / 200.0f);
}
```

Теперь все города растут при каждом вызове метода Update(), то есть нажатии на C. Но выглядят города всегда одинаково, поскольку изображаются с помощью звездочки, а по ней нельзя определить их размер. Это задача на пару минут, и вот как она выглядит на C#:

```
public override void Draw() {
    if (Size < 1000) {
        Console.ForegroundColor = ConsoleColor.DarkRed;
        Console.Write(".");
    } else if (Size < 10000) {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.Write(":");
    } else if (Size < 100000) {
        Console.ForegroundColor = ConsoleColor.DarkYellow;
        Console.Write("%");
    } else if (Size < 1000000) {
        Console.ForegroundColor = ConsoleColor.Yellow;
        Console.Write("*");
    } else if (Size < 10000000) {
        Console.ForegroundColor = ConsoleColor.DarkGreen;
        Console.Write("@");
    } else {
        Console.ForegroundColor = ConsoleColor.Green;
        Console.Write("#");
    }
}
```

Этот код отображает города различных размеров не только своим символом, но цветом, и размер города можно определить с первого взгляда.

Надеюсь, вы теперь осознали пользу объектной ориентированности. Мы переделали методы Update() и Draw() в классе SnivCity, значительно изменив то, что происходит в игре, причем пальцем не тронули главный класс Snivilization. Это называется инкапсуляцией: функциональность каждого объекта заключена внутри него, а не находится где-то снаружи, поэтому изменения этой функциональности отображаются везде. В примере со Snivilization, каждый город сам себя прорисовывает. Мы могли бы переместить этот код в главный цикл DrawScreen(), но вдруг нам понадобится нарисовать город где-либо еще, допустим, на каком-нибудь экране города? Потребуется скопировать код туда. Любые изменения в функциональности придется дублировать в разных участках кода, а это лишь увеличивает вероятность ошибки и замедляет код – куда лучше инкапсулировать функцию в Draw().

Инкапсулируемся глубже

Для разъяснения, как лучше использовать инкапсуляцию, я перестрою нашу игру, чтобы каждый город возвращал налоги и выручку от науч-

Скорая помощь

Закрытые переменные видны только внутри класса, которому они принадлежат, тогда как открытые переменные доступны вне класса. Есть также защищенные переменные, доступ к которым может осуществляться как из своего класса, так и из любых его классов-наследников.

ных исследований (и такая бывает!) в родительскую цивилизацию. Как и с прорисовкой, заставить класс Snivilization лезть в каждый SnivCity и вытаскивать налоговые суммы – не очень хорошая идея: мы ведь можем изменить алгоритм в любой момент, и неплохо бы централизовать код.

Сначала добавьте пять переменных сразу после объявления Rand в классе Snivilization:

```
public static int TaxRate = 8;
public static int ScienceRate = 25;
public int TotalCash;
public int CashLastTurn;
public int ScienceLastTurn;
```

Переменные TaxRate и ScienceRate статические, и они могут быть доступны напрямую через класс Snivilization. Например, должна ли налоговая ставка влиять на рост городов? Конечно, должна; поэтому изменим теперь метод Update() класса SnivCity, чтобы учесть налоговую ставку:

```
public override void Update() {
  float growthrate = 25;
  growthrate -= Snivilization.TaxRate;
  Size *= 1 + (growthrate / 200.0f);
}
```

Согласно этой формуле, высокие налоги заставляют города уменьшаться – так на самом деле и бывает! Мы также можем использовать ставки на налоги и науку, чтобы города сами рассчитывали свои налоги и прибыль от науки; кроме того, численность населения у нас рассчитывается с плавающей запятой – добавим метод GetSize(), преобразующий ее в целое число, пригодное для других методов. Вот как выглядит этот код:

```
public int GetTax() {
   return (int)Math.Round((Size / 400.0f) * Snivilization.TaxRate);
}
public int GetScience() {
   return (int)Math.Round((GetTax() / 100.0f) * Snivilization.
ScienceRate);
}
public int GetSize() {
   return (int)Math.Floor(Size);
}
```

Каждый город может теперь рассчитывать свои налоговые и научные показатели; у нас есть переменные, сохраняющие значения прибыли от науки и налоговых отчислений с прошлого хода, да еще и цикл foreach, обновляющий все города. Как же рассчитать полные показатели налогов и науки? Ответ: легко! Новый метод UpdateCities() должен очищать CashLastTurn и ScienceLastTrun, затем добавлять к этим значениям значения методов GetTax() и GetScience() для каждого города, например так:

```
public void UpdateCities() {
   CashLastTurn = 0;
   ScienceLastTurn = 0;
   foreach (SnivCity city in Cities) {
      city.Update();
      CashLastTurn += city.GetTax();
      ScienceLastTurn += city.GetScience();
   }
  TotalCash += CashLastTurn;
}
```

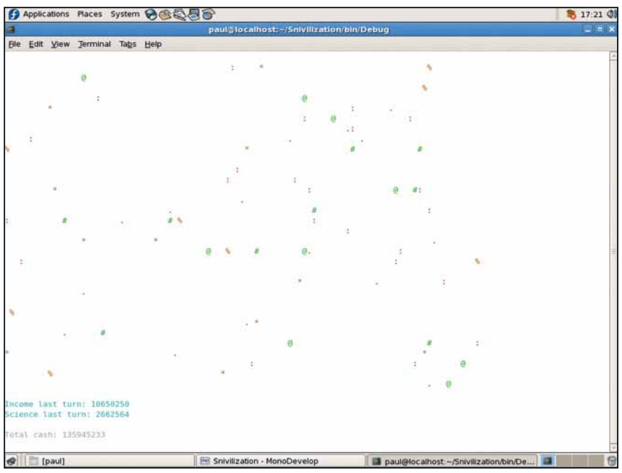
Эти числа можно использовать для получения информации о благосостоянии игрока. Пока что наш метод DrawScreen() просто рисует карту, но добавим в конец метода следующий код:

```
Console.WriteLine("");
Console.ForegroundColor = ConsoleColor.Cyan;
Console.WriteLine("Income last turn: " + CashLastTurn);
Console.WriteLine("Science last turn: " + ScienceLastTurn);
Console.WriteLine();
Console.ForegroundColor = ConsoleColor.White;
Console.WriteLine("Total cash: " + TotalCash);
```

и получим краткое описание статуса игры и карту. На диске вы обнаружите, что я расширил этот код так, что игрок может нажимать P, S, T и получать информацию о населении, науке и налогах для своих городов. С ростом городов код начисления налогов и успехов науки может изменяться, поэтому надо быть сумасшедшим, чтобы не оставить их в методах GetTax() и GetScience().

Хотя мы рассмотрели только функции, возвращающие значения, можно также использовать функции для установки переменных. Например, у вас есть переменная GovernmentType и вы установили ее в «Democracy», то наверняка захотите что-то изменить. Присваивание GovernmentType = Governments.Democracy не окажет мгновенного эффекта на игру. Но действие вроде SetGovernment(Goverments.Democracy) может меняться во времени по мере того, как города будут бунтовать, подвергаться нападению врагов и так далее.

Отведенное нам место исчерпано, но, надеюсь, вы убедились в пользе объектной ориентированности. Она помогает формировать «контракты» с вашим кодом, например, заставляя всех наследников SnivObject реализовывать Draw() и Render(). Она также позволяет добавлять объектам функциональности без копирования больших участков кода. Наш проект допускает множество вариантов его расширения. Интересуетесь ли вы интеллектом противника, графикой или поиском пути, дерзайте. Будете в городе, черкните мне пару строк о ваших успехах (**paul.hudson@futurenet.co.uk**)! LXF



› *Snivilization* **Пола Хадсона: на таком снимке экрана, вероятно, она практически невидима.**

Скорая помощь

Нажмите Ctrl+C, чтобы выйти из игры в любой момент. Возможно, вы захотите добавить что-нибудь более интеллектуальное – попробуйте вместо бесконечного цикла Run() использовать булеву переменную finished, устанавливающуюся в **true** при нажатии определенной клавиши.

Скорая помощь

В коде на нашем диске я добавил класс SnivUnit, чтобы показать, как несколько дочерних классов могут наследовать от одного базового класса. Однако сам SnivUnit помечен как абстрактный, потому что содержит абстрактный метод CanMove().

SnivUnit спроектирован как абстрактный класс, потому что нет такой сущности, как «боевая единица» – есть воздушные боевые единицы, морские боевые единицы, наземные и так далее. Чтобы создать морскую боевую единицу, нужен класс SnivSeaUnit, наследованный от SnivUnit, и метод CanMove(), возвращающий **true**, если клетка на карте является морем.

» **Через месяц** Используем двухъядерные процессоры для многопоточности!

Blender:

С помощью Python и Blender **Бен Харлинг** набросает среду для обмена сообщениями, открывая дорогу ко всестороннему взаимодействию.





Наш эксперт

Бен Харлинг
Бен работает штатным программистом, применяя Blender, Python и многие другие программы с открытым кодом в создании графических интерфейсов и другого ПО..

Есть отличный мультик, под названием Noob make MMORPG, сделанный парнем с Украины, который часто появляется на форумах *Ogre 3D*. Его можно найти вот здесь **http://snipurl.com/1eff4** Это история про начинающих разработчиков игр, которые, в эйфории от изобилия открытых решений для работы с графикой, хватаются за дело и сляпывают собственное жалкое подобие *World of Warcraft*. История способна сильно охладить энтузиазм новичков; и все-таки создать 3D-среду взаимодействия за очень короткий промежуток времени возможно – используя *Blender* совместно с Python.

На данном уроке мы разработаем простое приложение обмена сообщениями с 3D- интерфейсом на *Blender*, управляемое скриптом на Python и пригодное для обслуживания множества пользователей. Приложение предоставит большие возможности для расширения или включения его в ваш собственный проект *Blender*, а также может быть легко встроено в игру или же работать отдельно. Примечательно, что с этим приложением я сумел установить единую сессию обмена сообщениями между Windows, Linux, OS X и даже Playstation 3 с помощью одного и того же файла *Blender*, произвольно обрабатываемого с любой платформы – а это явно вызовет некоторое изумление.

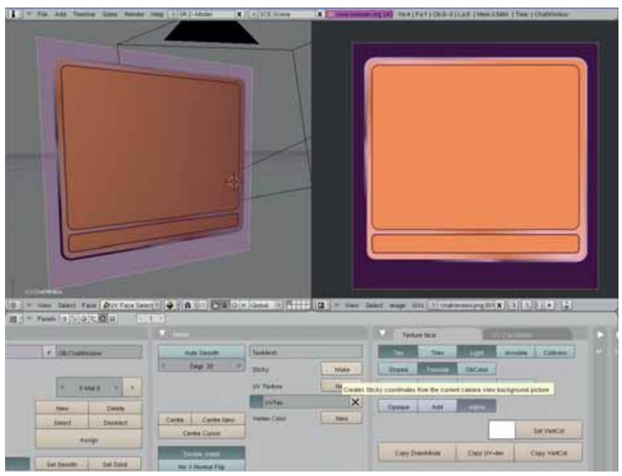
Чтобы добиться такого, нужны всего-навсего последние версии Python и *Blender* для вашей системы. Вам также потребуется нарисовать интерфейс пользователя для окна чата. Я использовал *Inkscape*, чтобы создать макет простого графического интерфейса с окном для текста и с полем ввода. В данном проекте понадобятся растровые шрифты для отображения текста в *Blender*, поэтому я предусмотрел их подборку на диске. Шрифты можно создать инструментом *FTBlender*, но на время написания этого руководства были доступны версии только под FreeBSD и Windows.

Создаем клиента и интерфейс

Запустите Blender и удалите куб или любой другой примитив, который по умолчанию появится в окне. В качестве окна мы собираемся использовать простую плоскость, поэтому создайте такую, развернув лицом к себе; выровняйте камеру и увеличьте изображение, пока оно не займет все окно. Затем разделите окно просмотра и измените одну из панелей на UV-редактор изображения (*UV Image Editor*). Вернитесь на вид с 3D и нажмите F для перехода к выбору граней UV.

Теперь войдите в редактор изображений, нажмите A, чтобы выбрать все вершины, затем нажмите Image > Open и загрузите свой интерфейс. Вызовите кнопки редактирования нажатием F9, и увидите панель Texture Face, которая активна, потому что мы все еще в режиме UV Face Select. В этой панели активируйте кнопки Tex, Light, TwoSide и Alpha. Кнопка Alpha полезна при использовании изображений с альфа-каналом прозрачности.

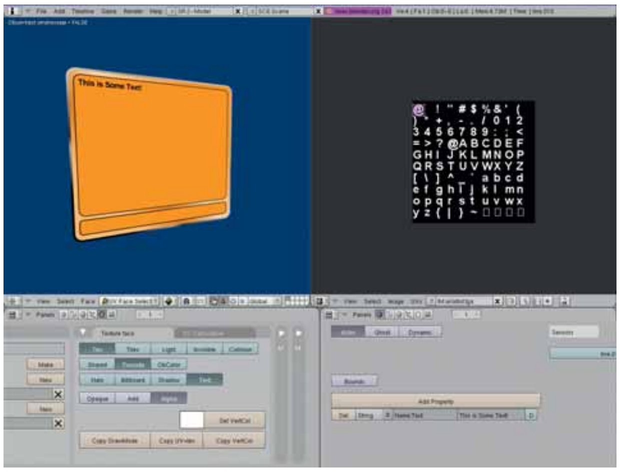
Перейдите обратно к 3D-виду и проверьте, что кнопка Viewport Shading установлена в режим Texture. Вы должны увидеть свой интерфейс отраженным на многоугольник. Если он развернут неправильно, перейдите обратно к UV Face Select, выберите снова все вершины UV, затем нажмите R для поворота. Чтобы повернуть на нужный угол, наберите значение в градусах (скажем, 90) и нажмите Enter. Выровняв текстуру, создайте новую небольшую плоскость, перекрывающую фон – проверьте, что она немного ближе к камере, чем большая плоскость. Поместите ее внизу интерфейса (она станет первой строкой текста) и еще раз с выбранной плоскостью установите 3D-окно в режим UV Face Select (нажмите F в окне 3D), затем загрузите растровый шрифт (например **arailbd.tga** с компакт-диска) в качестве текстуры для плоскости. На этот раз вам понадобится уменьшить масштаб вершин в редакторе изображений, чтобы они комфортно размещались рядом с символом @, который появляется на изображении шрифта (показан на панели справа, выше). Как и до этого, активируй-



› **Начало окна чата. Кнопки UV диктуют, как объект с текстурой будет выглядеть в движке игры.**

» **Месяц назад** Мы уменьшили головную боль от Java-разработки с помощью Ant.

Чат будет почат



› Нажмите P, переведя курсор на окно 3D-вида, для запуска игрового движка и проверки своей работы. Заметьте, что здесь выбран знак @.

те кнопки Texture Face, а также нажмите на кнопку Text в том же окне, чтобы *Blender* знал, как использовать объект для отображения текста. Переименуйте объект (не сетку!) во что-нибудь вроде txtLine.

Теперь нажмите F4 для активации панели Logic. Добавьте строковое свойство с именем Text к плоскости и задайте ему некое тестовое значение, например, This is some text! или что-нибудь столь же оригинальное. Перейдите к 3D-виду, нажмите клавишу P, и если все пойдет хорошо, вы увидите ваш текст поверх фона. Если текст имеет большие межбуквенные расстояния, уменьшите объект-плоскость в режиме Edit Mode (нажмите Tab в 3D-виде), потом сделайте предпросмотр, и так пока не будет достигнуто нужное расстояние; затем увеличьте масштаб в режиме Object Mode, чтобы восстановить размер шрифта.

Доведя до ума первую строку, скопируйте ее несколько раз, размещая строки друг над другом, чтобы заполнить главное графическое окно. Это гарантирует нам, что текст будет правильно отображаться при каждом новом сообщении. В моем примере у меня получилось 16 строк, но вы вольны сделать их сколько угодно; надо, однако, записать, сколько именно (не считая строки ввода), так как некоторые скрипты учитывают количество отображаемых строк.

Скопируйте текстовую плоскость и поместите ее над полем ввода вашего интерфейса. Переименуйте его как txtInput. Вам понадобится добавить свойство sendmessage под свойством Text для этой панели, показывающее, что это сообщение готовится к отправке. Наконец, создайте пустой объект в любом месте и назовите его controller.

Начинаем расширять

С пустым выбранным объектом переходим на панель Logic, добавляем сенсор Always и отключаем обе кнопки (pulse mode). Добавьте контроллер Python и свяжите сенсор с его вводом. Измените один из видов на текстовый редактор и добавьте новый скрипт client_connect. На свежесозданном контроллере Python есть кнопка Script. Нажмите ее и введите client_connect.

Теперь мы готовы начинать писать код на Python, чтобы вдохнуть жизнь в наше приложение. Все необходимые файлы есть на диске, и мы займемся только ключевыми моментами в коде, а за полными листингами обращайтесь к файлам. Обращаем внимание, что код, касающийся сети, в этом проекте не очень безопасен, поэтому испытания рекомендуем проводить в частной сети.

Наш контроллер, естественно, должен настроить приложение и попытаться подсоединиться к узлу, чтобы начать разговор с другими участниками. Сначала нам нужно импортировать игровой движок *Blender* (далее называемый ИД Blender) и стандартную библиотеку Python socket с помощью

```
import GameLogic as g
import socket
scn = g.getCurrentScene()
```

Последняя строка получает ссылку на текущую сцену, а заодно и на все объекты в ней. Следующие три строки определяют узел подключения, порт и отдельное имя каждого клиента. Вы можете обернуть эти свойства в графическое окно входа, используя те же методы, что были описаны выше. Далее мы начинаем назначать глобальные свойства, хитро присоединяя их к модулю GameLogic. Сложно точно сказать, когда скрипт будет выполнен в ИД Blender, поэтому простым способом избежать ошибок будет использование модуля движка для взаимодействия между скриптами. Мы будем использовать функцию hasattr для создания порядка выполнения. Она запрашивает объект на наличие определенного свойства GameLogic, затем создает его, если оно не существует, или совершает действия над ним, если существует.

Первое свойство, которое надо определить – это список плоскостей, созданных нами для отображения текста. Создадим его поиском в результатах вызова функции getObjectList() по ссылке на сцену: скрипт будет добавлять плоскости в порядке их создания, и если вы создавали их последовательно снизу вверх, то в списке mDisplay они окажутся в нужном порядке.

```
# загрузить объекты для отображения сообщений
if not hasattr(g, “mDisplay”):
        g.mDisplay = []
for obj in scn.getObjectList():
        if ‘line’ in obj.name:
                g.mDisplay.append(obj)
```

Далее мы попробуем подсоединиться к узлу и создадим свойство GameLogic, если соединение прошло удачно. Снова используем здесь hasattr – для гарантии, что мы не замещаем предыдущее соединение. Начнем с простого объекта socket и попробуем установить соединение. Хотя это не лучшее решение, мы обрамим все в конструкцию try…except, которая позволит функции «упасть» красивее. После установки соединения, немедленно пошлем серверу наше имя для записи в журнал. Наконец, установим флаг на GameLogic, сигнализируя, что соединение было успешно установлено, и остальные скрипты могут начать работу.

Контроллеру требуется установить еще один сенсор Always, чтобы »

LINUX FORMAT На DVD
» Blender 2.43
» Код примеров статьи

Добавьте чат в свои игры

Вы можете легко включить эту программу обмена сообщениями в качестве отдельной сцены во время игровой сессии и вызывать его на лету как отдельное меню. Чтобы это сделать, переименуйте сцену, содержащую обмен сообщениями, затем начните новую сцену, которая будет содержать вашу игру. Окно беседы может вызываться с помощью любой комбинации сенсора ввода, контроллера Always и актуатора Scene Actuator, установленного в Add Overlay Scene, все устанавливается на панели Logic и не требует написания скриптов. Добавьте все это в пустой объект в вашей игровой сцене для быстрого интегрирования чата. Вы можете также отделить скрипты клиента и соединения от программы чата, для более гибкого общения с сервером.

» он издавал сигнал True каждые пять тиков или около того. Он управляет частотой обновления клиента с сервера, и сильно влияет на ширину канала, которую пытается занять клиент. Данный проект не использует балансировку нагрузки, поэтому потребуется более подробная модель взаимодействия с сервером, если вы захотите использовать этот код в реальности. Снова, создайте новый Python-контроллер и ассоциируйте с ним новый текст скрипта. Назовите его client и добавьте следующие строки:

```
import GameLogic as g
import pickle
import socket

cont = g.getCurrentController()
own = cont.getOwner()
scn = g.getCurrentScene()

if not hasattr(g, 'message'):
  g.message = ""

if hasattr(g, "connected"):
  toSend = pickle.dumps(g.message, pos)
  g.conn.send(toSend)
  g.message = ""
  data = g.conn.recv(1024)
  if data:
    messages = pickle.loads(data)
    # display the messages list
    for n in range(len(g.mDisplay)):
      g.mDisplay[n].Text = messages[n]
```

Мы импортируем игровой движок и модуль socket, как и раньше, а также модуль Python pickle, для подготовки наших данных к передаче по сети. Далее, мы получаем ссылку на объект, к которому подсоединен скрипт, посредством ссылки на логический блок, представляющий данный Python-контроллер. Мы также добавляем еще одно свойство к GameLogic, которое будет содержать любой текст для отправки.

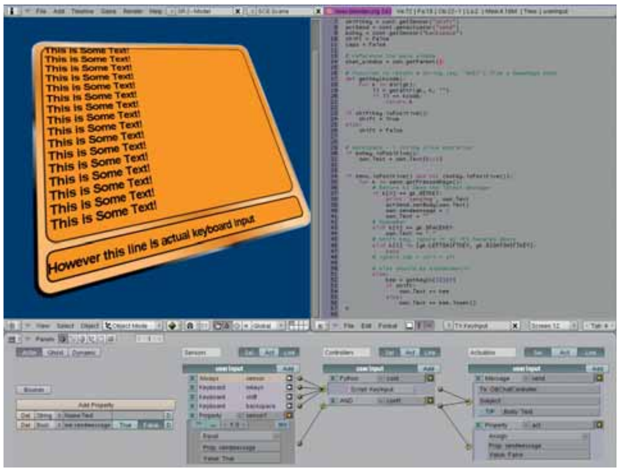
Остальная часть скрипта зависит от того, удачно ли произошло

«Я запускал единую сессию обмена сообщениями между Windows, Linux, OS X и даже PS3 одним файлом Blender.»

соединение с сервером. По каждому сигналу сервера клиент собирает все локальные данные – в нашем случае просто свойство Text входного объекта – и «закатывает» его в файлоподобный объект, пригодный для передачи. Затем данные пересылаются на сервер, а GameLogic.message сбрасывается. Клиент ожидает ответ от сервера, который, будучи полученным, «разворачивается» в список строк. Затем мы перебираем глобальный список текстовых панелей, определенный ранее, и назначаем каждой из них соответствующую строку.

Обработаем ввод

Далее нам необходимо присоединить новый скрипт к объекту ввода. Он будет обрабатывать ввод с клавиатуры и преобразовывать его в символы для отображения в нашем окне. Объекту ввода требуется сенсор приема клавиши, подсоединенный к новому Python-контроллеру и новому скрипту. ИД Blender включает модуль GameKeys, который можно использовать для преобразования кодов клавиш в полезные символы. Модуль GameKeys – это, попросту говоря, справочник имен и номеров кодов клавиш, поиск по которому осуществляется сравнительно легко. На самом деле, вы можете и не прибегать к скриптам, просто используя логические блоки, слушающие определенные клавиши и добавляющие символы в свойство Text. Но, скажем честно, вы сохраните кучу времени, используя Python в качестве решения, как и во многих других вещах.



› **Логические блоки настроить непросто, зато потом можно обойтись без кода.**

Со скриптом ввода клавиши все ясно, а вызов printdir(GameKeys) быстро выведет все коды клавиш, если у вас настроен консольный вывод. Я присоединил сенсоры, слушающие определенные клавиши, к тому же скрипту, чтобы продемонстрировать некоторые альтернативные установки. Скрипт получает ссылки на эти сенсоры и автоматически вызывается при нажатии определенной клавиши.

Скрипт ввода также связан и имеет ссылку на объект Message Actuator, именуемый send. При нажатии клавиши Enter скрипт устанавливает у актуатора свойство Body равным свойству ввода Text. Затем он изменяет свойство sendmessage на True, сигнализируя, что сообщение готово к отправке. Другой сенсор, на сей раз сенсор свойства, присоединен, чтобы слушать наличие True у sendmessage. Если это так, то запускается актуатор сообщения, и оно посылается контроллеру Empty. Мы добавим к контроллеру еще один сенсор, слушающий сообщение. Тут вызывается новый скрипт, просто вставляющий свойство GameLogic.message в тело получаемого сообщения. Сообщение уже автоматически отослано и очищается скриптом client каждые пять тиков, так что по части клиента у нас все в порядке.

Обслуживание клиентов

Наш сервер выполняет совершенно другую Python-программу, поэтому запустите вашу любимую IDE для работы с Python и создайте новый файл **chat_server.py**. Собственная версия этого файла опять-таки есть на диске, поэтому за полным кодом обращайтесь к нему. Наш сервер использует преимущества потоков для одновременного обслуживания множества клиентов, но помните, что приведенный пример примитивен, и его можно значительно улучшить.

После импорта требуемых модулей приступим к определению клиентского класса на стороне сервера. Он будет дублироваться при каждом новом соединении и работать одновременно вместе со своими собратьями:

```
# Наш класс потока:
class ClientThread ( threading.Thread ):
  # Создаем этого клиента и его поток
  def __init__ ( self, channel, details ):
    self.channel = channel
    self.details = details
    self.position = [0,0,0]
    self.message = ""
    self.newPosition = [0,0,0]
    self.newMessage = ""

    # слушаем имя клиента, которое нам передается
    self.name = channel.recv(1024)
```

ВАУ-фактор

Сознайтесь честно, вы ведь не отказались бы обладать своей собственной MMORPG. Это, конечно, золотое дно. *Blender* и Python предоставляют отличную возможность сделать первые шаги в этом прибыльном жанре, совмещая простую 3D-графику в реальном времени (хотя это отнюдь не полноценный движок) с мощными возможностями сетевого и алгоритмического программирования. Вообще-то, массивной *Eve Online* не было бы без Python. Почему бы не обмениваться большим количеством данных между клиентом и сервером? Можно передавать позиции, скорости и почти любую информацию, и вы легко приступите к созданию своих постоянных миров и пригласите участвовать друзей.



› Форумы *Eve Online* помогут игрокам под Linux и OS X.

```
        print self.name, 'connected'

        # вызываем конструктор суперкласса
        threading.Thread.__init__ ( self )
    def run ( self ):
        # пока клиент не подсоединен
        while True:
            try:
                # слушаем данные от клиента
                data = self.channel.recv(1024)
                if data:
                    # распаковываем данные
                    self.newMessage = pickle.loads(data)
                    # если получено сообщение, добавляем его в список
                    if not self.newMessage == "":
                        txtList.append(self.name + ':' + self.newMessage)
                # выбираем последние 16 сообщений, которые отошлем
обратно
                txtToSend = txtList[-16:]
                # упакуем их и отошлем
                toSend = pickle.dumps(txtToSend)
                self.channel.send(toSend)
            except:
                # нет, мы потеряли соединение
                self.channel.close()
                print 'Connection to', self.name, 'lost'
        return False
```

Клиентский поток постоянно работает в цикле в каждой сессии *Blender*, обмениваясь данными «туда-обратно» (регулируется сенсором, установленным в *Blender*). Сначала поток ожидает сигнала от клиента (помните, в нашем клиентском скрипте *Blender* первым действием является отсылка данных), и если получены какие-то данные, они распаковываются и проверяются. Пустые строки мы игнорируем: если этого не сделать, GUI быстро переполнится ими. Если у нас есть какой-то текст, предваряем его именем клиента и добавляем текст в стек. Затем мы обрезаем массив ([-16:]), беря последние 16 записей, упаковываем результат и, наконец, посылаем его клиенту, где он будет выведен на экран.

Последние строки скрипта сервера определяют бесконечный цикл, в котором сервер ожидает на определенном сокете и создает потоки клиентов при поступлении соединения.

Командир, запускай!

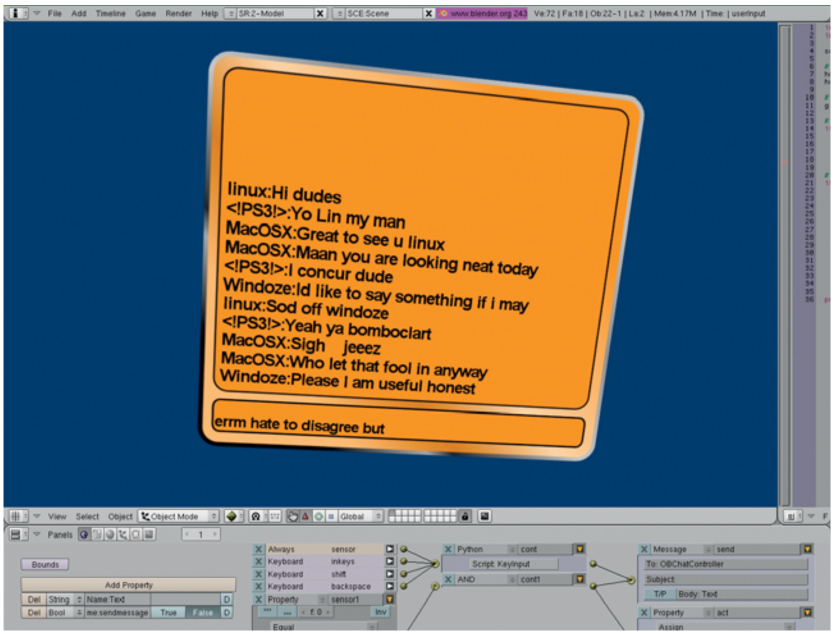
Программу обмена сообщениями легко протестировать локально; рекомендуем так и сделать, поскольку скрипт сервера выставит ваш компьютер напоказ в сети. Чтобы протестировать программу в оффлайне, установите свойство узла в localhost и используйте одинаковый номер порта для сервера и клиента. Может также оказаться, что сетевой экран блокирует соединения с локальной машиной, и потребуется разрешить входящие соединения для Python.

Сохраните вашу работу, затем откройте терминал, перейдите в каталог с проектом (cd) и наберите python chatserver.py для запуска. Когда он запустился, перейдите в *Blender* и нажмите P в 3D-виде. Если все в порядке, то тестовые значения, установленные на вашем дисплее будут заменены на пустые, загруженные с сервера. Напечатайте какой-нибудь текст, нажмите Enter, и он должен появиться в окне выше. Теперь запустите еще одну сессию *Blender*, нажмите P в этом окне, и магическим образом появится текст другой сессии!

Скрипт сервера можно запустить с любого сервера, на котором есть Python – просто установите адрес узла на ваших клиентах, и вы в игре. Уж не знаю, сколько клиентов выдержит такой сервер, но он вполне подходит для небольших переговоров. LXF

Скорая помощь

Мы включили несколько растровых .tga шрифтов на диск для этого проекта. Приложение *ftblender* может легко преобразовать шрифты TrueType в растровые, но оно пока существует только под Windows. *FTBlender* можно использовать только для преобразования шрифтов, не защищенных авторским правом.



› Итог проекта: несколько разных платформ общаются через *Blender*.

» **Через месяц** Натворим дел тэгом canvas из Firefox.

Anjuta: Создаем

ЧАСТЬ 7 Калькулятор, самая необходимая деталь рабочего стола, у нас уже есть – теперь настало время подумать и о текстовом редакторе. Берегись, Блокнот – **Андрей Боровский** запускает *Anjuta*!

– Разве имя должно что-то значить? – проговорила Алиса с сомнением.
– Конечно, должно, – ответил Шалтай-Болтай и фыркнул. – Возьмем, к примеру, мое имя – оно выражает мою суть!

Льюис Кэрролл, Алиса в Зазеркалье

В прошлой статье мы научились создавать программы GNOME своими руками, начиная, как говорится, с пустого файла. На этот раз мы автоматизируем процесс создания «заготовки» для нашей программы с помощью среды разработки *Anjuta* (в русском переводе, естественно, *Анюта*). Вот что написано о происхождении этого имени на сайте разработчиков *Anjuta* (**anjuta.sourceforge.net**): «[Это имя] не обозначает ничего, Наба Кумар (Naba Kumar) начал работу над проектом *Anjuta* и назвал его в честь своей девушки, так как программа посвящена именно ей». Для полного прояснения вопроса осталось узнать, что имели в виду родители, когда назвали свою дочь *Anjuta*...

Интегрированную среду *Anjuta* можно рассматривать как аналог *KDevelop* для *GTK+*. Помимо приложений *GTK+* и GNOME, *Anjuta* позволяет генерировать заготовки простых консольных программ, программ *wxWindows* (так это название пишется в настройках программы; ныне данный инструментарий называется *wxWidgets*), а также проекты «чистых» приложений X-Window, использующих для построения интерфейса только *Xlib*. В качестве языка программирования можно выбрать C или C++. Как и *KDevelop*, *Anjuta* не обладает собственными средствами визуального программирования – для этих целей используется уже знакомый нам *Glade*.

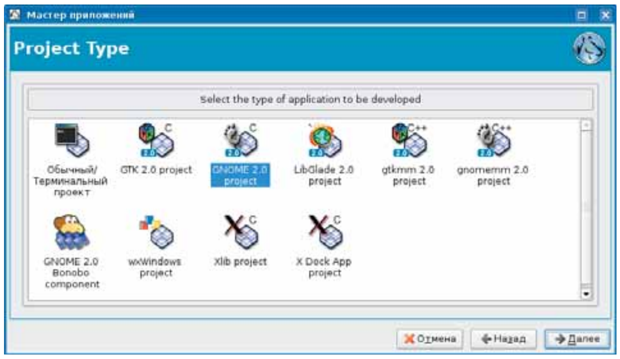
В этой статье мы детально исследуем процесс написания простенького текстового редактора GNOME средствами тандема *Anjuta* и *Glade*. Запустите программу *Anjuta*. По умолчанию, она открывает диалоговое окно (**Рис. 1**), в котором мы можем выбрать режим работы с *Anjuta*.



› Рис. 1. Окно выбора режима работы *Anjuta*.

Щелкните по кнопке Мастер приложений. Перед нами предстает первое окно Мастера приложений *Anjuta*.

Запустить Мастер приложений можно также с помощью команды меню Файл–>Новый проект... (команда Новый проект... расположена, почему-то в меню Файл», а не в меню Проект). В Мастере приложений нажмите кнопку Далее – вы перейдете к окну, в котором предлагается указать тип проекта (**Рис. 2**). Выберите проект GNOME 2.0. В ответ на щелчок по кнопке Далее открывается окно, в котором следует ввести название нового проекта, версию, имя автора и язык разработки (здесь следует выбрать C: если указать C++, вы лишитесь функции автоматической генерации кода с помощью *Glade*).



› Рис. 2. Окно выбора типа проекта *Anjuta*.

Еще одно важное окно Мастера приложений (**Рис. 3**) позволяет нам определить дополнительные параметры нового проекта – нужно ли добавлять текст лицензии GPL во все файлы исходных текстов и добавлять ли код, предназначенный для работы с *gettext*. Даже если вы не собираетесь выполнять интернационализацию своего приложения в обозримом будущем, ее поддержку все равно стоит включить, просто на всякий случай. Окно дополнительных параметров проекта позволит вам активировать режим генерации исходных текстов с помощью *Glade* (в *Glade 2.x* это все еще возможно), что и следует сделать. В этом же окне Мастера приложений можно выбрать пиктограмму для нового приложения и указать, нуждается ли оно для своей работы в открытом окне консоли. На этом генерация нового проекта заканчивается.

В левой части главного окна *Anjuta* расположено окно менеджера проектов. На вкладке «Проект» этого окна перечислены файлы исходных текстов приложения. Помимо файла **main.c** мы видим здесь файлы **callbacks.c**, **callbacks.h**, **interface.c**, **interface.h**, **support.c** и **support.h**. Если имена этих файлов показались вам знакомыми, интуиция вас не подвела – это файлы проекта *Glade* (мы недаром установили флажок Generate source code using Glade or Glademm в Мастере приложений). Теперь можно приступать к созданию интерфейса нашей программы.

Макет интерфейса

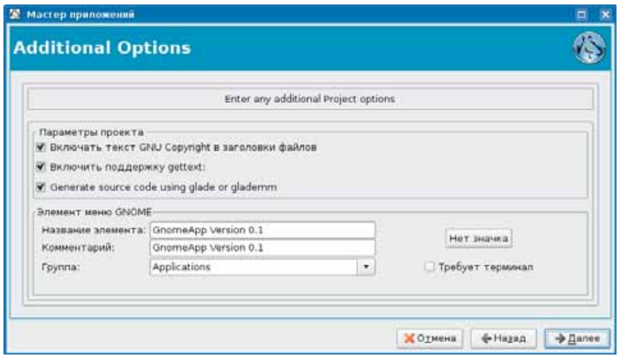
Команда меню Проект–>Редактировать графический интерфейс приложения... запускает экземпляр *Glade* для нашего проекта. При совмес-

Месяц назад Мы написали наше первое приложение GNOME – не используя при этом строку «Hello world».

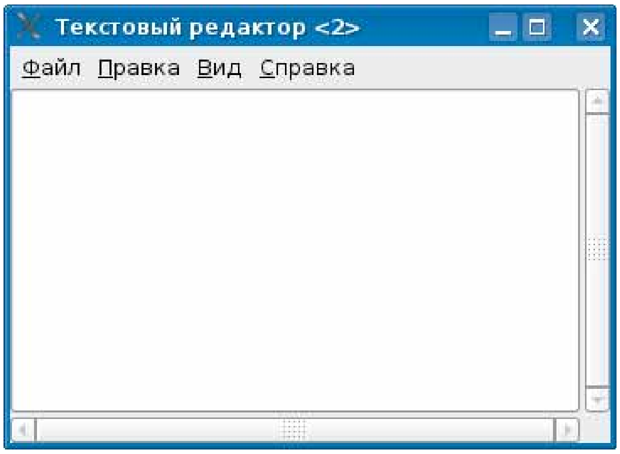
текстовый редактор

LINUX FORMAT На DVD

» Код примеров статьи



› **Рис. 3. Окно дополнительных параметров проекта.**



› **Рис. 4. Форма главного окна приложения.**

том использовании *Anjuta* и *Glade* следует учитывать, что эти программы слабо взаимодействуют между собой. Для того чтобы изменения, внесенные в проект с помощью *Glade*, стали известны *Anjuta*, нужно выполнить сохранение проекта в *Glade* (и не забывайте нажать кнопку Построить, чтобы *Glade* мог отразить в исходном коде изменения, внесенные в визуальном редакторе). Фактически, при работе с *Glade* нам почти всегда придется сохранять проект по два раза – один раз в *Glade*, другой раз – в *Anjuta*.

Получив доступ к окну визуального редактора *Glade*, мы можем воспользоваться нашими навыками проектирования интерфейсов *GTK+*. Откройте форму главного окна приложения (для этого, напомню, нужно дважды щелкнуть мышью по имени объекта-окна в главном окне *Glade*). Перейдите в редактор свойств *Glade* и назначьте свойству «Заголовок» объекта window1 строку «Текстовый редактор». Прежде чем добавлять в форму окна приложения GNOME какие-либо элементы управления, в нем следует разместить объект-контейнер. Мы воспользуемся контейнером типа «вертикальный бокс» с двумя строками. В верхнюю строку контейнера поместим визуальный элемент «Главное меню» (объект menubar1 класса GtkMenuBar). В нижнюю строку помещаем визуальный элемент «Просмотр текста» (объект textview1 класса GtkTextView). Теперь форма главного окна выглядит как настоящее окно текстового редактора (**Рис. 4**). Обратите внимание на то, что главное меню уже заполнено стандартными командами, такими как открытие и сохранение файла, копирование и вставка из буфера обмена и т.п. После добавления главного меню в форму окна приложения в файле **callbacks.c** появятся заготовки для обработчиков всех сигналов, генерируемых стандартными командами меню. Имена обработчиков сигналов команд меню создаются по шаблону on_menuitem_activate(), где menuitem – имя объекта, представляющего команду меню.

Выход здесь

Заполнение заготовки программы кодом мы начинаем, как и всегда, с кода завершения приложения. С помощью редактора свойств *Glade* назначьте обработчик сигналу destroy объекта window1 и добавьте в него вызов функции gtkmain_quit() (мы проделывали эту операцию уже много раз, так что останавливаться на ней подробно не будем). Приступим теперь к написанию обработчиков стандартных команд меню. Найдите обработчик on_quit1_activate() в файле **callbacks.c**. Он реагирует на сигнал, посылаемый командой меню **Файл | Выход**. Мы можем просто добавить в этот обработчик вызов on_window1_destroy():

```
void on_quit1_activate(GtkMenuItem *menuitem, gpointer user_data)
{
 on_window1_destroy(NULL, NULL);
}
```

Теперь вызов команды меню **Файл | Выход** приводит к завершению работы программы. Почему мы просто не добавили в обработчик on_quit1_activate() вызов функции gtkmain_quit()? Дело в том, что приложение GNOME (как и любое оконное приложение) может быть завершено несколькими разными командами. Удобно сделать так, чтобы при завершении работы приложения всегда вызвалась одна и та же функция (мы возлагаем эту роль на функцию on_window1_destroy()). В результате, если при завершении программы нам понадобится выполнять какие-то специальные действия (например, сохранять несохраненные данные), соответствующий код нужно будет добавить только в функцию on_window1_destroy(). Вместо непосредственного вызова функции on_window1_destroy() мы могли бы вызвать функцию gtk_widget_destroy(), передав ей в качестве аргумента указатель на объект window1. В следующей статье этой серии мы подробнее обсудим методы корректного завершения работы программы-редактора (в процессе вызова функции gtk_widget_destroy() генерируется сигнал destroy для соответствующего визуального элемента).

Теперь мы можем скомпилировать программу (сделайте это с помощью команды меню *Anjuta* Сборка–>Сборка). Перед тем, как выполнять компиляцию, следует сохранить все изменения, сделанные в исходном коде. Полные исходные тексты приложения-примера к этой статье вы найдете на диске. В процессе подготовки дистрибутива из него были удалены файлы сценариев сборки (не только make-файл, но и сценарий configure). При попытке выполнить команду сборки для этого дистрибутива вам будет сообщено об отсутствии make-файла. Для того, чтобы собрать программу из пакета, в котором отсутствуют файлы сценариев сборки, нужно сначала скомандовать Сборка–>Подготовить файлы для сборки....

Скажем несколько слов о редакторе исходных текстов *Anjuta*. В случае возникновения ошибки во время сборки редактор *Anjuta* выделяет подчеркиванием строку исходного текста, вызвавшую проблемы. В редакторе также присутствует рудиментарная функция автоматического завершения кода. В процессе ввода идентификатора, являющегося параметром функции, появляется список всех идентификаторов с подходящими именами, однако контекст при этом не учитывается, так что в списке идентификаторов, начинающихся на букву m, может оказаться функция main(). Если в текст программы добавляется вызов функции, определенной в файле проекта, редактор выводит подсказку со списком ее аргументов (к сожалению, она не работает для API- »

» функций *GTK+*/GNOME). Во всем остальном редактор исходных текстов *Anjuta* похож на *KWrite*.

Объекты GtkTextView и GtkTextBuffer

Теперь, когда мы научили программу корректно завершать свою работу, давайте заполним остальные обработчики сигналов меню. Начнем с обработчика сигнала, посылаемого командой **Файл | Новый** (функция on_new1_activate()). Прежде чем описывать команды, манипулирующие текстом, необходимо разобраться, как работает объект GtkTextView, который является основой нашего текстового редактора. Объект GtkTextView и связанный с ним объект GtkTextBuffer представляют собой чрезвычайно мощные средства, позволяющие создавать полноценные текстовые редакторы, поддерживающие разметку текста, вставку изображений, гиперссылок, сложную навигацию по тексту и т.п. В нашей программе мы воспользуемся лишь незначительной частью их возможностей. Заинтересованные читатели смогут узнать о них больше из справочной системы *GTK+*.

Всю сложную механику работы с текстом выполняет класс GtkTextBuffer. Объекты GtkTextBuffer содержат текст и вспомогательную информацию, необходимую для его отображения. Функции, работающие с GtkTextBuffer, позволяют выполнять различные операции с текстом. Объекты GtkTextView играют вспомогательную роль и служат, в основном, для отображения содержимого GtkTextBuffer на экране. Когда мы создаем экземпляр объекта GtkTextView, по умолчанию создается и экземпляр GtkTextBuffer. Однако текстовый буфер может существовать и отдельно от конкретного объекта GtkTextView. Один текстовый буфер может принадлежать нескольким объектам GtkTextView. В сложной программе, предназначенной для манипулирования текстом, вы можете даже создать несколько текстовых буферов для одного визуального элемента GtkTextView (несколько буферов можно использовать, например, для хранения промежуточных результатов редактирования текста) и делать их видимыми по мере необходимости.

Таким образом, для того, чтобы работать с текстом в нашем редакторе, мы должны получить доступ к текстовому буферу объекта GtkTextView. Вот как можно сделать это в обработчике on_new1_activate():

```
void on_new1_activate  (GtkMenuItem * menuitem, gpointer user_data)
{
  GtkTextView * textview1 = lookup_widget(menuitem, "textview1");
  GtkTextBuffer * textbuffer = gtk_text_view_get_buffer(textview1);
  gtk_text_buffer_set_text(textbuffer, "", 0);
  filename[0] = 0;
}
```

С помощью знакомой нам функции lookup_widget() (напомню, что ее, для нашего удобства, генерирует *Glade*) мы получаем указатель на объект GtkTextView, заданный своим именем. Далее мы получаем указатель на объект GtkTextBuffer с помощью функции gtk_text_view_get_buffer(). Задача нашего обработчика – очистить текстовый буфер. Для этого мы вызываем функцию gtk_text_buffer_set_text(). Эта функция, вообще говоря, предназначена для записи в буфер текста. Ее первым аргументом является указатель на объект GtkTextBuffer, второй аргумент представляет собой строку текста в кодировке UTF-8 (единственная кодировка, которую «признает» объект GtkTextBuffer). Третьим аргументом функции gtk_text_buffer_set_text() должна быть длина строки (в байтах, а не в символах). Мы записываем в буфер строку нулевой длины, что эквивалентно его очистке. Переменная filename объявлена глобально в файле **callbacks.c** как

```
static char filename[256];
```

и содержит текущее имя файла. В обработчике on_new1_activate() мы сбрасываем имя файла (если таковое было установлено). Следует отметить, что перед очисткой текстового буфера наш простой редактор не проверяет, содержит ли он несохраненные данные. Реализация такой проверки оставляется читателю в качестве домашнего упражнения.

В обработчике on_open1_activate(), который вызывается в ответ на сигнал, сгенерированный командой меню «Файл|Открыть», мы используем диалоговое окно GtkFileChooserDialog для выбора файла точно так же, как мы делали это в LXF90. Останавливаться подробно на коде этого обработчика мы не будем. Фактическая загрузка данных из выбранного файла выполняется с помощью определенной нами функции open_file(), которая, напротив, заслуживает более внимательного рассмотрения.

```
void open_file(GtkWidget * parentwindow, char * file_name)
{
        int fd, len;
        char * buf[1024];
        GtkTextView * textview1 = lookup_widget(parentwindow,
"textview1");
        GtkTextBuffer * textbuffer = gtk_text_view_get_
buffer(textview1);
        if ((fd = open(file_name, O_RDONLY, 0)) < 0)
        {
                GtkWidget * dialog;
                dialog = gtk_message_dialog_new(parentwindow,
GTK_DIALOG_DESTROY_WITH_PARENT,
GTK_MESSAGE_ERROR, GTK_BUTTONS_CLOSE,
"Невозможно открыть файл %s", file_name);
                gtk_dialog_run (GTK_DIALOG (dialog));
                gtk_widget_destroy (dialog);
                return;
        }
        strncpy(filename, file_name, 256);
        gtk_text_buffer_set_text(textbuffer, "", 0);
        while((len = read(fd, buf, 1024)) != 0)
                gtk_text_buffer_insert_at_cursor(textbuffer, buf,
len);
        close(fd);
}
```

Функция open_file() принимает два параметра. В первом передается указатель на объект, представляющий главное окно приложения. Он нужен нам, во-первых, для того, чтобы найти объект textview1, а во-вторых – чтобы вывести предупреждающее диалоговое окно в случае ошибки. Во втором параметре передается имя загружаемого файла.

В функции open_file() мы получаем указатель на текстовый буфер тем же способом, что и в функции on_new1_activate(). Далее мы пытаемся открыть файл, имя которого передано в параметре file_name, для чтения. Если открыть файл не удалось, выводится модальное диалоговое окно с информацией об ошибке.

Для вывода информирующих и предупреждающих сообщений в *GTK+* используется диалоговое окно GtkMessageDialog. Мы создаем экземпляр этого окна с помощью функции gtk_message_dialog_new(). Флаг GTK_DIALOG_DESTROY_WITH_PARENT указывает, что в случае уничтожения родительского визуального элемента диалоговое окно должно быть уничтожено вместе с ним. Флаг GTK_MESSAGE_ERROR указывает тип диалогового окна (окно с сообщением об ошибке). Для создания диалоговых окон других типов можно использовать флаги GTK_MESSAGE_INFO (нейтральное сообщение) GTK_MESSAGE_WARNING (предупреждение), GTK_MESSAGE_QUESTION (вопрос) и GTK_MESSAGE_OTHER (диалоговое окно неопределенного типа). Флаг GTK_BUTTONS_CLOSE указывает, что создаваемое диалоговое окно должно содержать одну кнопку, закрывающую окно. Если бы мы использовали

Еще раз о справке GTK+

Коль скоро речь зашла о справочной системе, нельзя не упомянуть о том, что браузер справки *Devhelp* не очень удобен. Страницы справки GTK+ обычно содержат довольно много текста, а в *Devhelp* (версии 0.1) отсутствует средство поиска текста на странице. Поскольку справка *GTK+* хранится в формате HTML, ее страницы можно читать непосредственно из каталога **/opt/gnome/share/gtk-doc/html**, однако при этом вы лишаетесь удобной возможности глобального поиска. Получить новейшую версию справки по *GTK+* можно на сайте **developer.gnome.org/doc/API**.

флаг GTK_BUTTONS_NONE, созданное нами диалоговое окно вообще не содержало бы кнопок. Флаг GTK_BUTTONS_OK_CANCEL позволяет создать окно с кнопками **OK** и **Cancel**. В справочной системе *GTK+* можно найти константы и для других комбинаций кнопок диалогового окна. Следующим параметром функции gtk_message_dialog_new() является строка форматирования текста, выводимого в диалоговом окне. Она может содержать форматирующие символы, точно так же, как и форматирующая строка функции printf(). Далее следуют аргументы, на которые ссылается форматирующая строка. Диалоговое окно становится видимым сразу же после вызова функции gtk_message_dialog_new(). По умолчанию функция gtk_message_dialog_new() создает немодальное окно. Хотя модальное окно можно создать, установив флаг GTK_DIALOG_MODAL, здесь мы поступаем иначе. Для того чтобы окно стало модальным (то есть, блокировало работу программы), мы вызываем функцию gtk_dialog_run(), которая приостанавливает выполнение программы до тех пор, пока не будет нажата одна из кнопок диалогового окна. Функция gtk_dialog_run() возвращает численное значение, соответствующее нажатой кнопке (у нашего окна только одна кнопка, так что значение, возвращенное gtk_dialog_run(), нас не интересует). После того как окно закрыто, мы уничтожаем соответствующий ему объект с помощью функции gtk_widget_destroy(). Если вы хотите вывести на экран немодальное окно и при этом получить информацию о том, какой кнопкой оно было закрыто, вы должны назначить обработчик сигнала response для объекта-окна.

После того, как мы убедились, что файл открыт, мы опустошаем текстовый буфер и начинаем заполнять его данными из файла. Функция gtk_text_buffer_insert_at_cursor() добавляет очередной блок текста в позиции курсора (по умолчанию – в конец текста). Таким образом, мы можем добавлять текст в буфер небольшими порциями (этот способ не самый изящный, но самый простой).

Функция on_save_as1_activate() обрабатывает сигнал, который создает команда меню **Файл | Сохранить как**. В этой функции мы также используем диалоговое окно GtkFileChooserDialog, только теперь оно настроено на выбор имени файла для сохранения. Обратите внимание на вызов функции gtk_file_chooser_set_do_overwrite_confirmation(). Будучи вызвана с аргументом TRUE, эта функция настраивает диалоговое окно выбора имени файла таким образом, что если файл с выбранным именем уже существует, программа спросит нас, действительно ли мы хотим перезаписать его. Сохранение данных в файле на диске выполняется с помощью функции save_file().

```
void save_file(GtkWidget * parentwindow, char * file_name)
{
        int fd, len;
        GtkTextIter start, end;
        gchar * buf;
        GtkTextView * textview1 = lookup_widget(parentwindow,
"textview1");
        GtkTextBuffer * textbuffer = gtk_text_view_get_
buffer(textview1);
        if ((fd = open(file_name, O_WRONLY|O_CREAT|O_TRUNC,
0666)) < 0)
        {
                GtkWidget * dialog;
                dialog = gtk_message_dialog_new(parentwindow,
GTK_DIALOG_DESTROY_WITH_PARENT,
GTK_MESSAGE_ERROR, GTK_BUTTONS_CLOSE,
"Невозможно открыть файл %s", file_name);
                gtk_dialog_run (GTK_DIALOG (dialog));
                gtk_widget_destroy (dialog);
                return;
        }
        gtk_text_buffer_get_start_iter(textbuffer, &start);
        gtk_text_buffer_get_end_iter(textbuffer, &end);
        buf = gtk_text_buffer_get_text(textbuffer, &start, &end,
FALSE);
        write(fd, buf, strlen(buf));
        close(fd);
        g_free(buf);
}
```

Список аргументов у этой функции такой же, как и у open_file(). Открытие файла (теперь, естественно, для записи) и проверка результата этой операции, с выводом диалогового окна в случае ошибки, выполняются практически так же, как и раньше. Самое интересное начинается потом. Мы получаем доступ к текстовому буферу объекта GtkTextView и должны извлечь из него текстовые данные для записи в файл. Получение текста из текстового буфера – совсем не такое простое дело, как может показаться на первый взгляд. Поскольку текстовый буфер GtkTextBuffer обладает чрезвычайно широкими возможностями в плане манипуляции текстом, он наделен довольно сложным интерфейсом. Интерфейс GtkTextBuffer упрощает выполнение сложных операций, но делает простые операции несколько более трудоемкими, по сравнению с аналогичными операциями в менее функциональных компонентах. Ключевую роль при обработке текста, содержащегося в буфере GtkTextBuffer, играет понятие итератора (их можно сравнить с итераторами в библиотеке шаблонов C++). Итератор текстового буфера *GTK+* представляет собой структуру типа GtkTextIter, которая указывает на некоторую позицию в тексте. Итераторы могут перемещаться по тексту в обоих направлениях в соответствии с заданным критерием. Например, с помощью итератора можно перейти к последнему символу текущей строки, найти следующее вхождение определенного символа или подстроки и т.п. Итераторы играют важную роль в поиске и навигации по тексту, но этим их функции не ограничиваются. Пара итераторов позволяет выделить фрагмент текста для дальнейших манипуляций с ним. Практически все функции текстового буфера, предназначенные для работы с фрагментами текста, используют итераторы.

Для того чтобы скопировать текст из буфера, мы определяем два итератора – start и end, которые должны указывать, соответственно, на начало и конец данных в буфере. Обратите внимание на то, что итераторы следует объявлять как переменные типа GtkTextIter, а не как указатели на этот тип, хотя функциям, работающим с итераторами, передаются именно указатели на соответствующие переменные. Функция gtk_text_buffer_get_start_iter() устанавливает переданный ей итератор в начале текста, содержащегося в буфере, а функция gtk_text_buffer_get_end_iter() – соответственно, в конце текста. Получив два итератора, указывающих на некоторый фрагмент текста (в нашем случае – на весь текст), мы можем скопировать его с помощью функции gtk_text_buffer_get_text(). Первым аргументом этой функции должен быть, как всегда, указатель на объект GtkTextBuffer. Второй и третий аргументы gtk_text_buffer_get_text() являются адресами переменных-итераторов, выделяющих фрагмент, который должен быть скопирован. Последний аргумент типа gboolean указывает, следует ли включать в копируемый текст невидимые символы (в нашем простеньком редакторе таковые отсутствуют). Функция gtk_text_buffer_get_text() возвращает строку с нулевым конечным символом, в которой содержится скопированный фрагмент текста. Для этой строки выделяется специальная область памяти, и по окончании работы с фрагментом текста строку нужно удалить с помощью функции g_free().

Если вы внимательно изучите прилагаемые к статье исходные тексты, то увидите, что мы пропустили некоторые функции нашего текстового редактора (например, работу с буфером обмена). Ему, а также другим интересным функциям *GTK+* и GNOME будет посвящена следующая статья. LXF

Через месяц Мы рассмотрим буфер обмена и другие «продвинутые» возможности GNOME.

Все на базу!

На DVD
- MySQL Connector/J 5.0
- commons-dbcp-1.2.2.jar и commons-pool-1.3.jar.
- Код примеров статьи

ЧАСТЬ 5 Ни одно серьезное приложение не обходится без большой базы данных. **Александр Бабаев** расскажет, что для этого требуется.

В предыдущих статьях цикла рассказывалось, как сделать интернет-приложение: правильно разделить создание интерфейса и бизнес-логику, а также реализовать некоторые технические детали. Но до сих пор мы использовали для хранения записей адресной книги простейшую сериализацию. Настало время сделать все «по-взрослому».

Хранение записей в базе данных имеет и достоинства, и недостатки. Из последних можно указать на необходимость поставки с приложением СУБД (системы управления базами данных), а также изучения специального языка запросов, который помогает сохранять данные в БД и получать их оттуда и достаточно сложен, если изучать его досконально.

Но есть и явные выгоды, иначе СУБД не использовались бы настолько широко, как они используются сейчас. Во-первых, язык, о котором я говорил (так называемый структурированный язык запросов, *Structured Query Language*, *SQL*) – более или менее стандартен. Это позволяет изучить его один раз, после чего использовать с разными СУБД. Эти системы также достаточно надежны и отлажены (и имеют приличную скорость работы), что позволяет использовать их в самых сложных ситуациях.

Хочется отметить еще одно свойство реляционных СУБД. Дело в том, что реляционными они называются не просто так. Есть специальная область математики, которая описывает такие объекты, как таблицы и поля, а также отношения (relations) между ними – реляционная алгебра и реляционное счисление. Знание основ этого раздела науки позволяет очень четко представить, что может и чего не может реляционная СУБД. А это, в свою очередь, дает понимание, где ее стоит, а где – не стоит использовать. В мире ПО это очень редкое и очень полезное свойство. Оно означает, что если сделать программу точно по математическим лекалам, то даже в условиях сверхдальнего космоса она будет работать так, как задумано. Значит, на нее можно положиться.

А теперь вернемся из космоса к нашим адресам…

JDBC

Рассмотрим, как можно обеспечить работу с СУБД, используя Java. Есть несколько подходов (например, применение специальных СУБД, которые имеют собственные API для Java), но в подавляющем большинстве случаев вам придется использовать *JDBC* (Java DataBase Connectivity). Это библиотека, которая позволяет работать с любой СУБД, для которой создан так называемый *JDBC*-драйвер. Он преобразует запросы *JDBC* в родные команды СУБД и обратно.

Мы будем использовать самую обычную связку: *MySQL* + стандартный драйвер *MySQL JDBC*. Если вы занимаетесь созданием интернет-приложений, то эта СУБД, скорее всего, у вас уже установлена. А драйвер можно скачать с сайта *MySQL* по адресу: http://dev.mysql.com/downloads/connector/j/5.0.html или найти на нашем DVD. Из всего архива нам понадобится только файл **mysql-connector-java-5.0.5-bin.jar**.

Работа с JDBC

Для выполнения дальнейших действий СУБД (*MySQL*d) должна быть запущена.

В целом, работа с *JDBC* не сильно отличается от работы с любым другим API (в PHP, например, все идеологически очень похоже). Во-первых, нужно подключиться к СУБД, т.е. создать соединение. После работы соединение обязательно нужно закрыть. Пока соединение «открыто», можно опрашивать СУБД, сохранять или получать данные, используя язык *SQL*.

Вспомним основные и самые простые команды *SQL*. Научимся создавать записи в БД, изменять и удалять их. Но прежде нужно создать саму базу данных.

Есть *MySQL* был только что установлен, то создать базу данных можно следующей командой:

```
mysqladmin -uroot create database addressbook
```

Теперь в этой БД нужно сделать таблицу, которую мы будем использовать для хранения записей. Сделаем маленькую программу на Java, которая выполнит эту работу за нас (**см. листинг 1**). Заодно посмотрим на общую структуру процесса работы с СУБД.

Листинг 1. Создание таблицы

```
import java.sql.*;

public class CreateTable {
```

Месяц назад Мы разобрались с паттерном Model-View-Controller (MVC).

```
  public static void main(String[] args) throws Exception {
    // эта строчка загружает драйвер MySQL, чтобы его можно было использовать
    Class.forName("org.gjt.mm.mysql.Driver").newInstance();
    // получим соединение
    Connection connection = DriverManager.
        getConnection("jdbc:mysql://localhost:3306/addressbook", "root", "");

    try {
      // создадим «выражение»
      Statement statement = connection.createStatement();
      statement.executeUpdate(
          "CREATE TABLE `addresses` (\n" +
          "`name` varchar(200) default NULL,\n" +
          "`number` varchar(200) default NULL\n" +
          "`comment` varchar(200) default NULL,\n" +
          ") DEFAULT CHARSET=utf8");
      statement.close();
    } finally {
      // закроем соединение
      connection.close();
    }
  }
}
```

В этой программе присутствует одна «волшебная» строчка, которая загружает драйвер. Нужно отметить, что в Java 6 она необязательна. Остальное более или менее понятно. Мы открываем соединение (легко видеть, что при этом мы подключаемся к локальному компьютеру, порту 3306 и базе данных addressbook, с именем пользователя root и пустым паролем), потом генерируем выражение (statement) и выполняем *SQL*-код, который и создает таблицу. Затем выражение «закрывается», после чего в final-блоке закрывается и само соединение. Final-блок нужен для того, чтобы соединение было закрыто независимо от возникновения исключения в блоке try ... catch. Как правило, число одновременных соединений ограничено, а значит, это ценный ресурс, которым нужно пользоваться аккуратно.

Вместо кода, который создает таблицу, можно написать любой другой *SQL*-запрос, например, добавляющий запись или изменяющий ее. Для получения записи нужен несколько другой код, и его мы обсудим чуть позже.

Чтобы добавить запись в уже созданную таблицу, можно использовать следующий код (**См. листинг 2**):

Листинг 2. Создание записи в таблице

```
private void addEntry(String aName, String aNumber, String aComment)
throws SQLException {
  Connection connection = DriverManager.
  getConnection("jdbc:mysql://localhost:3306/addressbook", "root", "");

  try {
    PreparedStatement statement = connection.prepareStatement(
        NSERT INTO `addresses` (`name`, `number`, `comment`) VALUES (?, ?, ?)");
    statement.setString(1, aName);
    statement.setString(2, aNumber);
    statement.setString(3, aComment);
    statement.executeUpdate();
    statement.close();
  } finally {
    connection.close();
  }
}
```

Чтобы поменять номер, следует воспользоваться командой UPDATE:

```
PreparedStatement statement = connection.prepareStatement(
        "UPDATE `addresses` SET `number`=?, `comment`=? WHERE `name`=?;
```

Внимательный читатель может заметить, что здесь используется не Statement, а PreparedStatement (предварительно подготовленное выражение). Оба они служат одной цели – с их помощью формируются и выполняются запросы к СУБД. Но Statement формирует запросы и тут же выполняет их, в то время как PreparedStatement сначала подготавливает запрос, затем заполняет его параметры значениями (места вставки параметров – это вопросительные знаки), и только потом выполняет его. PreparedStatement имеет множество преимуществ перед обычными запросами. Во-первых, предварительно подготовленные запросы могут кэшироваться прямо в СУБД. Это ускоряет их выполнение. Кроме того, можно выполнить несколько запросов сразу, используя пакетную обработку. Например, если нужно добавить не одну, а несколько записей, можно поступить примерно так:

- Создать PreparedStatement.
- Заполнить его (statement.set…).
- Вызывать statement.addBatch().
- При необходимости повторить предыдущие два пункта.
- Вызывать statement.executeBatch().

Теперь давайте разберемся, как можно вытащить информацию из БД. Это не намного сложнее (**см. листинг 3**).

Листинг 3. Получение записи из таблицы

```
Contact result = null;
PreparedStatement statement = connection.prepareStatement(
        "SELECT `name`, `number`, `comment` FROM `addresses` WHERE `name`=?");
statement.setString(1, aName);
ResultSet resultSet = statement.executeQuery();

if (resultSet.next()) {
    result.setData(resultSet.getString(1), resultSet.getString(2), resultSet.getString(3));
}
```

Теперь мы вызываем метод statement.executeQuery(), который возвращает resultSet. Согласно документации, он никогда не равен null, поэтому проверку я не произвожу. Далее вызывается метод resultSet.next(), который возвращает **true** тогда и только тогда, когда в результате есть еще одна запись, и, одновременно с этим, переходит на эту запись. После чего можно получить доступ к полям записи. Я получаю доступ по номерам, мне так проще. Но можно написать и resultSet.getString("name"), это тоже будет работать.

Но это будет работать плохо

Почему? Потому что никто в «нормальных» системах не работает с соединениями напрямую, а если работает, то никогда не открывает/закрывает соединение на каждый запрос. Иногда это может занять секунды и минуты. А сам запрос – несколько миллисекунд. Становится ясно, что создание соединения и его закрытие – узкое место, с которым нужно что-то делать. И решение называется «пул соединений». Пул (Pool) – это общий паттерн (см. LXF92), который состоит в следующем:

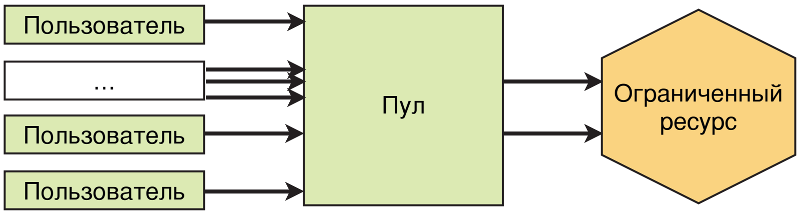
- У нас есть хранилище объектов. Представим себе оружейный ящик местной шпаны. В нем лежат несколько рогаток. Это и есть хранилище или пул.
- В каждый момент времени любой (кому позволено) может взять рогатку, попользоваться и положить обратно.
- Если рогатки нет, можно подождать, пока кто-нибудь другой не вернет ее в ящик.

Тогда, если время пользования рогаткой невелико (никто не держит ее долго), один пул может обслуживать очень большое количество хулиганов. Схематично пул изображен на рисунке. Видно, что при помощи небольшого количества соединений с ограниченным ресурсом обрабатывается большое количество «пользователей».

То же самое можно проделать и с соединениями. Создается

Скорая помощь

Напомню, что для того, чтобы подключить jar-файл к проекту, нужно указать в параметрах javac и java «-cp [имя jar-файлов через разделитель]», где разделителем является точка с запятой в Windows и двоеточие в Unix.



› Рис. 1. Схема работы пула.

А дайте нам ваше соединение!

Существует еще один способ работы с соединениями, который используется в крупных приложениях. Точнее, не в крупных, а в тех, где за соединениями следит контейнер приложений или контейнер сервлетов.

Для этого в дескрипторе нужно описать способ подключения к СУБД и параметры подключения. И при запуске контейнера можно будет получить DataSource, который будет проинициализирован самим контейнером. Как это делается – лучше смотреть в документации к контейнеру/СУБД. Например, для используемого нами Tomcat нужная информация находится тут: http://tomcat.apache.org/tomcat-5.5-doc/jndi-datasource-examples-howto.html.

пул, в котором держится открытыми, например, десять соединений. Если нужно произвести операцию с БД, соединение берется из пула, используется и возвращается на место. При этом оно остается открытым. И получается (кроме самого первого раза) – тоже уже открытым. Экономия зачастую просто огромна.

Полезные «Commons»

Если взглянуть на сайт http://jakarta.apache.org/, то можно увидеть очень много проектов, которые сделаны для Java. Среди них достаточно много хороших. Нам понадобится Commons (http://jakarta.apache.org/commons/). Это проект, в котором собраны маленькие «общие» библиотеки, которые приходится писать каждому программисту. Вместо того, чтобы терять время зря, можно просто скачать нужный пакет и использовать его. Сегодня нам будут интересны *commons-dbcp* и *commons-pool*. Их можно загрузить со страниц http://jakarta.apache.org/site/downloads/downloads_commons-dbcp.cgi и http://jakarta.apache.org/site/downloads/downloads_commons-pool.cgi соответственно или взять с нашего DVD. После скачивания нужно вытащить из архивов файлы **commons-dbcp-1.2.2.jar** и **commons-pool-1.3.jar**.

Основное предназначение второй библиотеки – создание пулов, первой – пулов соединений с СУБД. Давайте посмотрим, как все это используется. Для начала пул нужно инициализировать. Для этого мы напишем следующий метод:

```
String connectionString = "jdbc:mysql://localhost:3306/addressbook?user
=root&passowrd=";

GenericObjectPool connectionPool = new GenericObjectPool(null);
ConnectionFactory connectionFactory =
new DriverManagerConnectionFactory(connectionString, null);
new PoolableConnectionFactory(connectionFactory, connectionPool, null,
null, false, true);
PoolingDataSource _poolingDataSource = new PoolingDataSource(conne
ctionPool);
```

Теперь вместо создания соединения «самим» можно получать его из PoolingDataSource. Сама работа с соединением при этом не изменится ни на грамм. Вот, например, добавление записи (**листинг 4**, сравните с **листингом 2**):

Листинг 4. Добавление записи (используем пул соединений)

```
private void addEntry(String aName, String aNumber, String aComment)
throws SQLException {
 Connection connection = _poolingDataSource.getConnection();

 try {
   PreparedStatement statement = connection.prepareStatement(
     INSERT INTO `addresses` (`name`, `number`, `comment`) VALUES (?,
?, ?)");
   statement.setString(1, aName);
   statement.setString(2, aNumber);
   statement.setString(3, aComment);
   statement.executeUpdate();
   statement.close();
 } finally {
   connection.close();
 }
}
```

Повторяющийся текст оставлен специально. Чтобы было видно, что не поменялось ровно ничего. Даже close нужен. Но в действительности close перехватывается пулом соединений и вместо реального закрытия соединения возвращает его в пул.

Стоит упомянуть, что в данном случае используется шаблон Proxy. То есть создается класс, который унаследован от соединения, но в котором изменено поведение одного из методов (в данном случае, close). Благодаря этому код приложения не изменяется, но зато кардинальным образом изменяется поведение.

Другие пути

JDBC – великолепный способ работы с базами данных. Но не единственный. Главная особенность *JDBC* в том, что это самая низкоуровневая библиотека для работы с базами данных в Java. Все распространенные средства (EJB или Hibernate например), внутри так или иначе используют *JDBC*. Поэтому для понимания того, что происходит, *JDBC* знать просто обязательно. Иначе трудности отладки и недоумения по поводу «а что это тут произошло» будут преследовать вас постоянно и на каждом шагу.

Достоинства *JDBC* очевидны. Это сравнительная простота, мощность (мы не обсудили и десятой части того, что может эта библиотека: здесь есть и транзакции, и курсоры, и блокировки, и много чего еще), а также скорость работы. Но иногда требуется немного другое. В нашем случае в БД была только одна таблица, которая к тому же имела простую структуру. В сложных приложениях часто приходится работать с сотнями «бизнес-объектов», у которых бывает очень нетривиальная структура и взаимосвязи. При этом структура СУБД получается очень непростая. И процедура вытаскивания объектов (которые часто «раскиданы» по десяткам таблиц) – превращается в кошмар.

На этом этапе на передний план выходят другие библиотеки. В первую очередь это *Hibernate*. Во вторую – EJB. Во вторую, потому что EJB – это не только способ хранения объектов, но несколько более сложная «структура». А *Hibernate* – это типичный пример ORM (Object-Relational Mapping, объектно-реляционное отображение), то есть библиотеки, автоматически преобразующей объекты в строки таблиц и обратно. Работа с *Hibernate* сильно отличается от работы с *JDBC*, но внутри – это примерно то же, что мы и рассмотрели.

Также нельзя забывать, что несмотря на то, что *JDBC* – это унифицированная библиотека, которая работает с десятками различных СУБД (был бы драйвер), код, написанный для одной СУБД, далеко не всегда будет работать с другой. Разница обычно заключается в двух вещах:

» **Возможности драйвера**. Иногда что-то не реализовано. Например, вам нужны транзакции, а в драйвере их просто нет.

» **Разница в диалекте *SQL***. Ни одна СУБД не повторяет никакую другу

в языке запросов. Всегда есть различия. Поэтому, несмотря на стандарты, одно приложение не обязательно будет работать со всеми СУБД.

Ну вот, собственно, и все. Адресная книга успешно «покладена» в базу данных. Это отличное начало, но пора двигаться дальше. Интересно, а что же такое AJAX? LXF

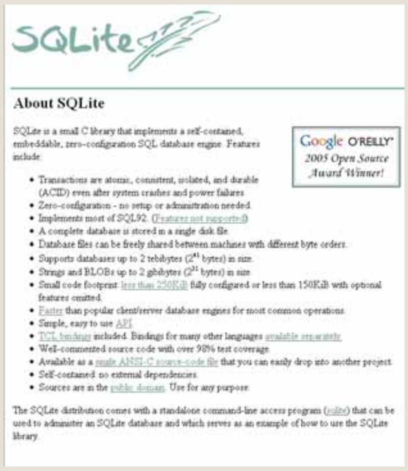
Что за штука... встраиваемые СУБД?

В Java 6 появилось новшество – некая Java DB. Что это? Это — встраиваемая СУБД (embeddable DB engine).

Обычно сервер СУБД работает отдельно (часто даже на своем собственном компьютере), но на этапе отладки приложения это зачастую не нужно. Да и в простых приложениях хочется, с одной стороны, использовать мощь баз данных, а с другой, распространять с программой сервер, учить пользователей им пользоваться – это кошмар.

Поэтому существует специальный класс СУБД, которые запускаются вместе с приложением (например, как обычный поток внутри вашей программы), и заканчивают работу вместе с ним. Их использование не отличается (за исключением общих разнящихся правил *SQL*-запросов) от работы с обычной СУБД: здесь тоже нужен драйвер (всё в поставке), те же выражения… Они сродни Jetty, которую мы использовали в самом начале, чтобы сделать наш первый сервлет.

Из известных встраиваемых СУБД можно назвать Java DB (это порт *Apache Derby*) и HSQLDB, используемую в *OpenOffice.org*, а из известных за пределами мира Java — *SQLite* (в принципе тоже подключаемую к Java).


SQLite

About SQLite

SQLite is a small C library that implements a self-contained, embeddable, zero-configuration SQL database engine. Features include:

Google O'REILLY 2005 Open Source Award Winner!

- Transactions are atomic, consistent, isolated, and durable (ACID) even after system crashes and power failures.
- Zero-configuration - no setup or administration needed.
- Implements most of SQL92. (Features not supported)
- A complete database is stored in a single disk file.
- Database files can be freely shared between machines with different byte orders.
- Supports databases up to 2 tebibytes (2^{41} bytes) in size.
- Strings and BLOBs up to 2 gibibytes (2^{31} bytes) in size.
- Small code footprint: less than 250KiB fully configured or less than 150KiB with optional features omitted.
- Faster than popular client/server database engines for most common operations.
- Simple, easy to use API.
- TCL bindings included. Bindings for many other languages available separately.
- Well-commented source code with over 98% test coverage.
- Available as a single ANSI-C source-code file that you can easily drop into another project.
- Self-contained: no external dependencies.
- Sources are in the public domain. Use for any purpose.

The SQLite distribution comes with a standalone command-line access program (sqlite) that can be used to administer an SQLite database and which serves as an example of how to use the SQLite library.

Через месяц Web 2.0 и AJAX – что означают нашумевшие маркетинговые словечки для мира Java?

Всё

ЧАСТЬ 10 Таблица – один из самых сложных текстовых элементов. Почему? Давайте спросим у **Евгения Балдина**!

Классификация химических элементов, позволяющая выявить зависимость их различных свойств от атомной массы.

Таблица Менделеева — пожалуй, самая полезная из таблиц.

Таблица – это особая форма передачи информации: далеко не всякое содержание следует облекать в такую форму. Но для малых объёмов однородных значимых данных таблицы зачастую оказываются наилучшим способом представления. Давайте разберемся, что предлагает в этой области *LaTeX*.

Немного теории

Содержимое таблицы организуется в колонки (графы) и горизонтальные строки таким образом, что каждый элемент является составной частью и строки, и колонки. Таблица состоит из следующих основных элементов: нумерационного и тематического заголовков (номера таблицы и её названия), головки[1] (заголовочной части таблицы), хвоста (всей остальной части, кроме головки), боковика (первой слева графы таблицы) и прографки (хвостовой части таблицы без боковика).

Боковик	Прографка		
ИЛИ	Истина	Ложь	**Головка**
Истина	Истина	Истина	**Хвост**
Ложь	Истина	Ложь	

› **Рис. 1. Структура таблицы.**

Заполняя таблицу текстовыми или цифровыми данными, полезно следовать некоторым правилам:

- Выносить общие для каждого элемента графы данные в её заголовок, а общие для строки данные – в заголовок боковика.
- По возможности, употреблять числа не более чем из четырёх значащих цифр. Более многозначные числа следует округлять. Общий множитель следует выносить в заголовок. То же самое нужно сделать и с единицами измерения.
- Перед знаком, отделяющим целую часть числа от дробной, всегда должна быть цифра. То есть, 0.1 – это правильно, а .1 – нет.
- Вместо отсутствующих данных проставлять многоточие …, Нет свед. и т.п. Если данных в принципе быть не может, то лучше отметить это с помощью тире: –.
- Не следует использовать знаки, означающее «то же, что и предыдущее значение». Лучше повторить число.

[1] В «издательском словаре-справочнике» А.Э. Мильчина утверждается, что попытка заменить этот термин по причине его «жаргонности» не удалась.

Честно говоря, описание таблиц с помощью *LaTeX*-разметки может показаться не очень удобным. Действительно, когда данные идут в перемешку с управляющими структурами, результат может выглядеть не очень красиво. Спасает только то, что большие таблицы с однородной структурой можно создавать с помощью скрипта, а маленькие таблицы не представляют особой проблемы. Как правило, поиск и проверка данных отнимает куда больше времени, чем их оформление.

tabbing

Если точно известны ширина столбцов, а таблица относительно простая, то можно воспользоваться окружением tabbing.

```
\begin{tabbing}
MMMMM  \= MMMMM \=      \kill
\textbf{\ ИЛИ} \> Истина \> Ложь  \\
Истина \> Истина \> Истина \\
Ложь   \> Истина \> Ложь
\end{tabbing}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

› **(Рис. 2).**

Первая строчка устанавливает положение табуляторов с помощью команды \=. Команда \kill, завершающая управляющую строку, даёт понять текстовому процессору, что её не надо печатать. Далее идёт обычный текст, где переход к следующей табуляции осуществляется с помощью команды \>, а перевод строки завершается комбинацией \\. Всё просто – точно так же таблицы печатали с помощью обычной печатной машинки, только табуляция задавалась «железным», а не программным способом.

К использованию окружения tabbing следует подходить с известной долей осторожности. Окружение формирует абзац, состоящий из отдельных строк, в котором нет места переносам и многострочным элементам. Зато это позволяет *LaTeX* легко переносить часть таблицы на следующую страницу. Абсолютно всё приходится делать своими руками, но в этом есть своеобразная прелесть. Зачастую tabbing становится базой для определения других, более специализированных окружений.

Внутри tabbing переопределяются некоторые стандартные команды *LaTeX* . Это, в частности, касается команд переноса (\-) и акцентирования (\' и \`). Для сохранения/воспроизводства текущей табуляции следует воспользоваться командами \pushtabs и \poptabs.

tabular и array

Самым популярным окружением для создания таблиц в *LaTeX* является tabular. Окружение array фактически полностью повторяет функци-

Месяц назад Мы научились создавать оглавления, предметные указатели и прочие «книжности».

о таблицах

онал tubular, но в отличие от последнего работает в математической моде – это полезно для создания матриц, которые, по своей сути, есть обычные таблицы.

```
\centering
\begin{tabular}[c]{l|ll}
\textbf{\ ИЛИ} & Истина & Ложь \\[2mm]\hline
Истина & Истина & Истина \\
Ложь   & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

› **Рис. 3.**

Данные делятся на ячейки с помощью символа «амперсанд» – &. Переход на следующую строку осуществляется стандартной командой переноса: \\. В качестве необязательного параметра ей можно указать дополнительный сдвиг по вертикали. Вертикальный размер каждой из строк автоматически выставляется в соответствии с высотой текста.

tabular создаёт единый объект – таблицу. Как и в случае картинок, таблицу удобно заключать в плавающее окружение table:

```
\begin{table}[ht]
 \centering %центрируем таблицу
  \begin{tabular}[«позиционирование таблицы»]{«формат столбцов»}
    «тело таблицы»
  \end{tabular}
 \caption{«подпись»}\label{tab:metka}
\end{table}
```

Это позволяет автоматически создавать нумерационный заголовок таблицы и добавлять тематический. Необязательный аргумент окружения tabular задает позиционирование всей таблицу по вертикали по отношению к окружающему тексту:

- t – выравнивание по верхней строке, то есть верхняя строка таблицы будет расположена на одном уровне со строкой, в которой она была размещена,
- c – выравнивание по центру,
- b – выравнивание по нижней строке.

Ширина столбцов вычисляется автоматически по формату, который задаётся через обязательный аргумент окружения. Каждому из столбцов должна соответствовать своя буква:

- l (left) – выравнивание по левому краю,
- c (center) – выравнивание по центру,
- r (right) – выравнивание по правому краю,
- p{«ширина»} – создание колонки определённой ширины. В случае жёстко заданной длины слишком большой текст может разбиваться на несколько строк.

Если формат столбцов повторяется, то для сокращения записи можно воспользоваться следующей спецификацией:

```
*{n}{«формат столбца или столбцов»}
```

где n – число повторений. Это своеобразный цикл.

```
\centering
\begin{tabular}{|*{3}{c|}}
\textbf{ИЛИ} & Истина & Ложь \\[2mm]
 \hline\hline
Истина & Истина & Истина \\
Ложь   & Истина & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь	Истина	Ложь

› **Рис. 4.**

Разделительные линии между столбцами создаются с помощью вертикальной черты |. Две вертикальные линии || формируют двойной разграничитель. Горизонтальные линии создаются с помощью команды \hline. По аналогии с двойной вертикальной чертой, две команды формируют двойную горизонтальную линию. Инструкция @{} позволяет вставить между столбцами любой символ, указанный в качестве обязательного аргумента. При этом подавляются околостолбцовые промежутки, добавляемые автоматически. Это можно быть полезно в случае, если один столбец представляет из себя какую-то измеренную величину, а второй её ошибку – тогда вместо разделительной черты между ними лучше поставить знак ±.

LINUX FORMAT На DVD
» TeX Live 2007

```
\centering
\begin{tabular}{c||p{2cm}@{$\pm$\ }r|}
\textbf{ИЛИ} & Истина & Ложь \\[2mm]
 \cline{2-3}
Истина & Истина & Истина \\
\cline{1-1}\cline{3-3}
Ложь    & Истина & Ложь\\ \cline{2-2}
\end{tabular}
```

ИЛИ	Истина	± Ложь
Истина	Истина	± Истина
Ложь	Истина	± Ложь

› **Рис. 5.**

Для того, чтобы «отчеркнуть» только часть столбцов, можно воспользоваться командой \cline{диапазон столбцов}.

В дополнение к стандартным типам столбцов, используемым в tabular, окружение array добавляет два новых:

- m{«ширина»} – то же, что и p{«ширина»}, но добавляется вертикальное выравнивание содержимого по центру клетки,
- b{«ширина»} – то же, что и m{«ширина»}, но вертикальное выравнивание содержимого происходит по нижней базовой линии последней строки.

Более подробную информацию об использовании окружения array можно найти в файле документации **array.pdf** из пакета *tools*.

Подробнее о том, как должна выглядеть таблица и как этого достичь, можно узнать в статье «Publication quality tables in *LaTeX*» (**booktabs.pdf**), написанной Саймоном Фиром (Simon Fear) для пакета *booktabs*. Этот пакет для тех, кто любит везде наводить лоск.

К вопросу о разделительных линиях

По характеру оформления линейки таблицы бывают закрытые (глухие), полузакрытые, открытые. Пользуясь случаем, настоятельно рекомендую:

- Никогда не использовать вертикальные линии в таблице;
- Иметь в виду, что двойные линии в оформлении таблицы также излишни.

Это, конечно, рекомендации, а не технические ограничения пакета, но если данные в таблице настолько разные, что хочется разделить их вертикальной линией, то проще сделать две таблицы.

Стиль hhline из коллекции *tools* определяет команду \hhline, которая позволяет создавать двойную рамку без ненужных пересечений. Подробности можно найти в документации: **hhline.pdf**.

Пакет *arydshln* необходим в случае использования пунктирных разделительных линий. Более чем исчерпывающая документация к этому пакету находится в файле **arydshln-man.pdf**.

» Клетки

Для объединения расположенных рядом (по горизонтали клеток) можно воспользоваться командой \multicolumn{n}{формат колонки}{текст}:

```
\centering
\begin{tabular}{|*{3}{c|}}
\textbf{ИЛИ} & Истина & Ложь \\[2mm]
 \hline\hline
Истина & \multicolumn{2}{c}{Истина} \\
Ложь   & Истина  & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	
Ложь	Истина	Ложь

› Рис. 6.

Первый обязательный параметр соответствует числу объединяемых колонок, второй – формату результирующей колонки, третий – это текст.

Для объединения клеток по вертикали можно воспользоваться пакетом *multirow*, в котором определяется набор одноимённых команд:

```
\centering
\begin{tabular}{|*{3}{p{1.2cm}|}}
\textbf{ИЛИ} & Истина & Ложь \\[2mm]
 \hline\hline
Истина & \multirow{2}{1.2cm}{Истина} & Истина \\
Ложь   &                              & Ложь
\end{tabular}
```

ИЛИ	Истина	Ложь
Истина	Истина	Истина
Ложь		Ложь

› Рис. 7.

которые принимают два или три обязательных параметра:

\multirow{«число строк»}{«ширина»}{«текст»}

\multirow{«число строк»}*{«текст»}

В качестве первого аргумента передаётся число строк, которые займёт объединённая клетка, далее можно указать ширину ячейки или включить ее её автоматическое вычисление. Кстати, число строк может быть и отрицательным. В этом случае объединяются клетки, расположенные выше команды. Подобное может потребоваться, чтобы согласовать свою работу с пакетом *colortbl*:

```
\centering
\begin{tabular}{c>{\columncolor{yellow}}cc}
\backslashbox{два}{один} & Истина & Ложь \\
\rowcolor{yellow}
Истина &                         & Истина \\
Ложь   & \multirow{-2}*{Истина} & Ложь
\end{tabular}
```

два \ один	Истина	Ложь
Истина	Истина	Истина
Ложь		Ложь

› Рис. 8.

Пакет *colortbl* предназначен для раскрашивания таблицы в разные цвета. Принципы работы пакета подробно излагаются в файле документации **colortbl.pdf**.

Команда \backslashbox{текст}{текст}, делящая клетку на две части наклонной чертой, определена в пакете *slashbox*. Иногда так оформляют клетку на перекрестии боковика и головки. С пакетом идёт пример: **slashbox.tex**.

Автор кириллических шрифтов **lh** Ольга Лапко предлагает для более изощрённой работы с таблицами пакет *makecell*. В нем определена команда \makecell, которая «создаёт окружение одноколоночной таблицы с предопределёнными общими параметрами выключки, интерлиньяжа и вертикальных отбивок вокруг. Её удобно использовать для многострочных ячеек. Дополнительный аргумент команды позволяет изменить выключку таблицы». В пакете содержится документация на английском языке (**makecell.pdf**), которая дублируется русскоязычной (**makecell-rus.tex**).

Выравнивание чисел

Часто содержимым таблицы являются некоторые числа, поэтому повышенное внимание к их выравниванию (с целью упрощения восприятия) является вполне оправданным. Пакет *dcolumn* из коллекции *tools* добавляет ещё одну спецификацию к формату столбцов таблицы:

```
\centering
\begin{tabular}{|c|l|D{.}{,}{5}|}
0.3141 & 0.3141 & 0.3141 \\
3.141  & 3.141  & 3.141 \\
31.415 & 31.415 & 31.41 \\
\end{tabular}
```

0.3141	0.3141	0,3141
3.141	3.141	3,141
31.415	31.415	31,41

› Рис. 9.

Новая спецификация имеет формат: D{delim}{output}{nfruc}, где delim – символ или набор символов, по которому происходит выравнивание (обычно это точка или запятая), output – символ, который замещает delim при компиляции (например, иногда бывает нужно заменить точку на запятую), nfrac – максимальное число позиций в дробной части числа (при отрицательном значении число позиций не фиксируется). Дополнительную информацию можно найти в документации: **dcolumn.pdf**.

Пакет *rccol* обладает схожей функциональностью, но умеет округлять значения. К сожалению, мне не удалось заставить его нормально работать в случае, когда в качестве разделителя используется точка. Возможно, это просто недоразумение – по крайней мере, документация (**rccol.pdf**) утверждает, что с точкой все должно быть в порядке.

Пакет *warpcol* предоставляет общую процедуру форматирования числовых колонок. В его документации (**warpcol.pdf**) представлены примеры, иллюстрирующие различные эффекты, связанные с выравниванием.

Доступ к данным

CSV (от англ. Comma-Separated Values – значения, разделённые запятыми) – это текстовый формат, предназначенный для представления табличных данных. Для прямого доступа к ним можно воспользоваться пакетом *csvtools*. Документация (**csvtools.pdf**) достаточно подробна, но лучше все же написать скрипт, который будет считывать необходимые сведения и генерировать готовый tex-файл.

Клоны tabular

tabularx из коллекции *tools* – это расширение *tabular*, позволяющее автоматически вычислять ширину колонок, имеющих форматный определитель *X*.

```
\centering
\begin{tabularx}{\textwidth}{|D{.}{,}{4}|X|c|}
0.3141 & очень длинная строка  & 0.3141 \\
3.141\footnote{В окружение tabularx можно
добавлять подстрочное примечание}
 & 3.141  & 3.141
\end{tabularx}
```

0,3141	очень длинная строка	0.3141
3,141[a]	3.141	3.141

[a] В окружение tabularx можно добавлять подстрочное примечание

› Рис. 10.

В качестве первого аргумента окружения *tabularx* передаётся ширина таблицы. При компиляции *X* преобразуется в p{ширина}, где вместо ширины подставляется величина, автоматически вычисленная таким образом, чтобы ширина таблицы в целом оставалась неизменной. Если в таблице присутствует более одного столбца, имеющего формат *X*, то для ее формирования может потребоваться несколько проходов. Все подробности, естественно, можно почерпнуть из документации (**tabularx.pdf**).

Пакет *tabulary* также является модификацией стиля *tabular*, а точнее – *array*. Он был создан для автоматического расчёта ширины колонок таблицы с целью минимизации ее высоты. Детали можно найти в документации: **tabulary.pdf**. Пользоваться одноимённым окружением следует с некоторой долей осторожности.

ctable предлагает альтернативный подход к оформлению таблицы. Вместо окружения используется команда \ctable, которая объединяет в себе функциональность *tabular* и *table*. В дополнение к этим особенностям в \ctable можно вставлять команды создания подстрочных примечаний для таблицы. Примеры и документацию к пакету можно найти в файле **ctable.pdf**.

Многополосные таблицы

Окружение tabular и его производные хороши всем, за исключением того, что они не могут занимать больше одной страницы. Для вёрстки на несколько страниц обычно используют одно из двух окружений: supertabular и longtable, которые определённы в одноимённых стилях. Они обладают схожей функциональностью, но разными способами оформления. Как supertabular, так и longtable предоставляют возможность создания стандартной шапки и стандартного окончания, появляющихся в начале (на каждой новой странице) и в конце таблицы. Подробности можно отыскать в документации: **supertabular.pdf** для supertabular и **longtable.pdf** для longtable из коллекции *tools*.

В отличии от supertabular, longtable гарантирует неизменность ширины столбцов на протяжении всей таблицы – это достигается несколькими проходами *LaTeX*. Неизменность ширины столбцов в зависимости от ситуации может рассматриваться и как преимущество, и как недостаток.

Кроме окружения supertabular, в одноименном пакете определено окружение mpsupertabular, которое заключает каждую отдельную страницу («кусок») в окружение minipage, что позволяет использовать подстрочные примечания прямо в таблице.

Более молодым и, возможно, правильным является пакет *xtab*. По сути дела, это расширение *supertabular* с некоторыми улучшениями и исправлениями. Подробная документация доступна в файле **xtab.pdf**.

Вывод

Вывод – таблица без линеек или организованный в колонки и строки материал, который не разделён линейками. Оглавление \tableofcontents, списки иллюстраций \listoffigures и таблиц \listoftables тоже являются таблицами.

```
\centering
\begin{tabular}{p{0.7\textwidth}@{}r}
Глава 1\dotfill & 2 \\
Глава 2\dotfill & 10 \\
Глава 3\dotfill & 124
\end{tabular}
```

Глава 1	2
Глава 2	10
Глава 3	124

› **Рис. 11.**

Команда \dotfill формирует отточия (заполнение пространства точками).

Окружение listliketab из одноимённого пакета специализируется на создании таблиц, оформленных как перечисление. Это окружение будет очень кстати при оформлении вопросников:

```
\storestyleof{enumerate}
\begin{listliketab}
 \newcounter{tabenum}\setcounter{tabenum}{0}
 \newcommand{\nextnum}{\addtocounter{tabenum}{1}\thetabenum.}
 \begin{tabular}{L>{\bf}l@{~~или~~}>{\bf}l@{~~или~~}>{\bf}l}
 \nextnum & Красный  & зелёный & голубой \\
 \nextnum & Низкий & средний & высокий \\
 \nextnum & Радостный & грустный  & удивлённый \\
 \end{tabular}
\end{listliketab}
```

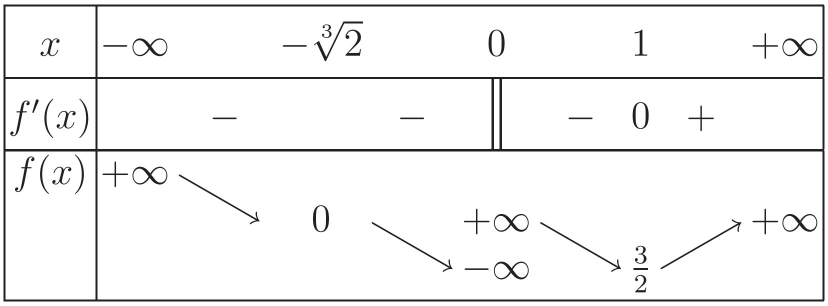
1.	**Красный**	или **зелёный**	или **голубой**
2.	**Низкий**	или **средний**	или **высокий**
3.	**Радостный**	или **грустный**	или **удивлённый**

› **Рис. 12.**

Документация с примерами находится в файле **listliketab.pdf**.

И это тоже таблицы?

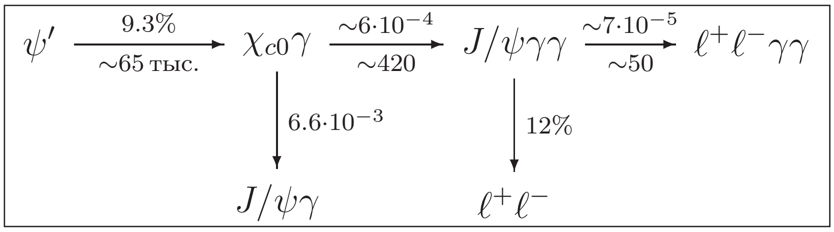
Пакеты *tableaux* и *tabvar* созданы для исследования области определения и поведения функции. Пригодится при изучении или написании методички по началам матанализа. Молодцы французы!



› **Рис. 13.**

tabvar является более современной версией стиля tableaux. Поэтому подробности и примеры лучше всего искать в его документации: **tabvar.pdf** и **demo.tex**.

Пакет *pb-diagram* пригодится, когда нужно нарисовать простенькую диаграмму (это, конечно, не совсем таблицы, но определенное сходство наличествует):



› **Рис. 14.**

Завершая разговор о таблицах

Таблица – это один из самых сложных текстовых элементов. Сложных не потому, что его трудно оформить – трудно добыть подходящие данные. Сделать бессмысленную таблицу очень легко: обилие информации ненадолго задержится в памяти читателя. Но думать о том, как и что представлять, всегда полезно: хорошая таблица зачастую стоит тысячи слов. LXF

» **Через месяц** Завершая этот длинный цикл, мы познакомимся с программированием в *LaTeX*.

Новая серия! Разбираемся с электронной подписью и учимся использовать ее для защиты персональных документов

Электронные подписи

ЧАСТЬ 1 В наши дни никому нельзя доверять. Милая девушка из чата может легко оказаться бородатым мужчиной, а его сообщения – искусно подделанными зеленым человечком с Альфы Центавра. Избежать конфуза поможет электронная подпись – и **Андрей Боровский** сейчас расскажет, как именно.

– Что ж, теперь, когда мы увидели друг друга, – сказал Единорог, – можем договориться: если ты будешь верить в меня, я буду верить в тебя! Идет?
– Да, если вам угодно, – отвечала Алиса.
Льюис Кэрролл, Алиса в Зазеркалье

Жизнь научных идей так же непредсказуема, как и человеческая. Мог ли сэр Эрнст Резерфорд в начале XX века представить себе, что через каких-то пятьдесят лет профессия исследователя атомного ядра станет престижной и опасной? Точно так же математики прошлого, изучавшие такие проблемы алгебры, как разложение чисел на простые множители, не могли предположить, что сфера их профессиональных интересов когда-нибудь привлечет к себе внимание большого бизнеса, спецслужб и международного криминала.

Широкое распространение компьютеров в повседневной жизни привело к не менее широкому распространению систем шифрования и электронной подписи. Сегодня уже все знают, что электронная криптография предназначена не только для разведчиков и дипломатов. Не все, однако, представляют себе, как пользоваться даже самыми простыми средствами защиты электронных документов, такими как цифровые сертификаты и удостоверяемые ими ключи. На сегодняшний день наиболее широкое распространение получила технология удостоверения электронной подписи на основе системы PKI (Public Key Infrastructure, Инфраструктура открытых ключей). В мире Linux (да и не только) также набирает популярность система цифровых подписей на основе GnuPG. Однако, то, что просто и удобно в Linux, может быть совсем не простым в Windows с ее рудиментарной консолью. Если вы хотите обмениваться безопасной почтой и документами с пользователями Windws, у вас есть два варианта действий – убедить их установить консольные программы безопасности, разработанные в соответствии с философией Linux, или воспользоваться сертификатами PKI (их еще называют сертификатами X.509, по названию используемого протокола). В этой статье речь пойдет именно о них. Администраторы web-серверов также имеют дело с сертификатами PKI, в основном, в связи с протоколом SSL. Мы поговорим о сертификатах с точки зрения «рядового пользователя ПК». Чтобы получить какую-либо пользу от сертификатов, даже рядовому пользователю необходимо знать, как они работают.

О терминологии

Поскольку в этой статье речь пойдет о вещах, связанных с шифрованием, нельзя не коснуться вопроса о криптографической терминологии. Разные отечественные авторы применяют разную терминологию, а многочисленная переводная литература по шифрованию, над которой трудятся переводчики разной квалификации и прилежания, только усиливает путаницу. Набора терминов, который бы нравился всем, не существует. Термины, используемые в этой статье, взяты из «Краткого словаря понятий, сокращений и терминов PGP, криптографии и информационной безопасности», расположенного на сайте http://www.pgpru.com/. Всех, кто желает прояснить для себя, чем расшифровка отличается от дешифровки и что такое хеш-функция, равно как и всех, недовольных моей терминологией, я отсылаю к нему.

Алиса, Боб и все-все-все

Для правильного понимания работы цифровых сертификатов необходимо иметь, по крайней мере, общее представление о важной отрасли современной криптографии – шифровании с открытым ключом (известным так же как «асимметричное шифрование»). В ассиметричном шифровании участвуют два ключа. То, что зашифровано одним ключом, может быть расшифровано с помощью другого ключа (в некоторых алгоритмах, например, в популярном RSA, верно и обратное, то есть выбор ключа для шифрования и расшифровывания может быть произвольным, однако не все алгоритмы обладают этим свойством). Важной особенностью алгоритмов асимметричного шифрования является то, что, зная один ключ, невозможно вычислить другой (так, по крайней мере, должно быть). Что это дает пользователям компьютеров? Предположим, что некий пользователь (согласно одной из традиций, существующих в криптографической литературе, назовем этого пользователя Алисой) желает получать конфиденциальные сообщения. С помощью программы асимметричного шифрования Алиса может создать пару ключей. Один из этих ключей (секретный ключ, СК) Алиса будет хранить в строжайшей тайне, а второй ключ (открытый ключ, ОК) опубликует на своей страничке в Интернете. Если другой пользователь, Боб, захочет послать Алисе секретное письмо, он зашифрует его с помощью ОК Алисы. Такое письмо можно будет расшифровать только с помощью СК Алисы и никак иначе. Преимущество асимметричной схемы заключается в том, что Бобу не нужно встречаться с Алисой для обмена ключами. Даже если они раньше не были знакомы, они могут начать конфиденциальную переписку (разумеется, для того, чтобы Боб мог принимать конфиденциальные сообщения по этой схеме, у него должна быть своя пара ключей СК/ОК). Впрочем, на практике асимметричные алгоритмы не используются напрямую для шифрования сообщений.

и цифровые сертификаты

Как уже отмечалось, в некоторых алгоритмах ОК и СК взаимозаменяемы (иными словами, то, что зашифровано с помощью СК, может быть расшифровано с помощью ОК и никак иначе). Взаимозаменяемость ключей позволяет использовать такие алгоритмы не только для шифрования, но и для электронной подписи. Предположим, что Алиса посылает Бобу письмо, зашифрованное ее секретным ключом с помощью алгоритма RSA. Поскольку это письмо можно расшифровать только с помощью ОК Алисы, Боб может удостовериться, что автором письма является обладатель СК Алисы (то есть, с высокой степенью вероятности, сама Алиса), а не кто-то другой. Это и есть простейший вариант цифровой подписи. Заметим, что само сообщение, зашифрованное с помощью СК Алисы, не является секретным (ведь ОК Алисы доступен всем). Подпись удостоверяет только подлинность сообщения.

Поскольку цифровая подпись предназначена не для обеспечения конфиденциальности, а для подтверждения подлинности сообщения, в шифровании всего сообщения нет нужды. Вместо этого можно создать цифровой отпечаток сообщения – некоторое количество битов (меньшее, чем количество битов в сообщении), полученное таким образом, что любое изменение в сообщении приведет к изменению отпечатка (такие отпечатки создаются с помощью однонаправленных хэш-функций, поэтому их называют хэш-значениями). Для того, чтобы подписать свое письмо, Алиса получает его хэш-значение с помощью одного из стандартных алгоритмов, шифрует это хэш-значение своим СК и прикрепляет зашифрованное хэш-значение к своему сообщению. Для того, чтобы удостовериться в авторстве сообщения, Боб должен расшифровать его хэш-значение, используя ОК Алисы, затем сравнить расшифрованное значение со значением, вычисленным для сообщения с помощью того же алгоритма. Если какой-нибудь злоумышленник, скажем, Зина (или Зиновий, кому как нравится) изменит сообщение Алисы, Боб обнаружит это, сравнив зашифрованное Алисой хэш-значение с тем, которое он вычислил (изменить электронную подпись Зина не сможет, если только не знает СК Алисы). Следует признать, что на практике все выглядит не так просто. Поскольку длина хэш-значения значительно меньше длины сообщения (в этом и заключается смысл использования хэш-значений), одному хэш-значению соответствует более чем одно сообщение. Теоретически это означает, что Зина может подобрать другое сообщение, имеющее то же хэш-значение, что и сообщение Алисы, прикрепить к этому сообщению подпись Алисы и отправить его Бобу. Получив такое сообщение, Боб будет уверен, что его автор – Алиса. Однако, поскольку хэш-функция, с помощью которой получают хэш-значение сообщения, необратима, подобрать документ с тем же хэш-значением можно только путем слепого перебора (что обычно очень трудно сделать). Впрочем, разработчики криптографических систем – тоже люди, и главное отличие человека от Бога – способность ошибаться – присуще им, как и всем остальным. На протяжении долгого времени для получения цифровых подписей использовался алгоритм хэширования MD5, однако не так давно было обнаружено, что он ненадежен. Оказалось, что во многих случаях подобрать текст под заданное значение MD5, даже на не самом мощном ПК, можно за несколько минут. Сейчас алгоритм MD5 выводится из употребления и заменяется алгоритмом SHA1, который, пока что, считается надежным.

По сравнению с обычными «бумажными», у электронных подписей есть и достоинства и недостатки. Важнейшим недостатком электронной подписи является то, что злоумышленник, завладевший СК Алисы, сможет подписывать документы от имени Алисы, и никто, включая саму Алису, не сможет опровергнуть авторство документа. Из этого факта вытекает одно интересное следствие. Зная, что ее секретный ключ может, в принципе, попасть в чужие руки, Алиса имеет возможность воспользоваться этим фактом в собственных бесчестных целях, а именно – подписать некий документ, а затем заявить, что она этого не делала (а секретный ключ у нее украли). Использование меток времени, надежно удостоверяющих, когда именно документ был подписан, позволяет затруднить этот вид мошенничества, известный как «отказ от подписи», но умная Алиса почти всегда сможет подгадать время «потери» ключа так, что уличить ее будет невозможно. Хорошая новость, связанная с использованием электронных подписей, заключается в том, что эти подписи действительно защищают весь документ, ведь изменение хотя бы одного бита в документе приведет к изменению его цифрового отпечатка. В этом смысле цифровая подпись намного надежнее «бумажной», которая зачастую ставится на одной-единственной странице многостраничного документа.

В схеме электронной подписи сообщений с помощью асимметричного шифрования возникает одна проблема, для решения которой и предназначены цифровые сертификаты. Боб может доверять электронной подписи Алисы, только если он уверен, что ОК, который он использует для расшифровки хэш-значения, действительно принадлежит Алисе. Если ОК опубликован на домашней страничке Алисы, то, скорее всего, он действительно принадлежит ей. Но страничку Алисы могут взломать злоумышленники, другие злоумышленники могут открыть «псевдо-сайт» Алисы и выложить на нем свой ключ, и это не говоря уже о том, что у Алисы может просто не быть своей домашней странички. Для надежной связи с Алисой Бобу необходимо какое-то удостоверение подлинности Алисиного ОК. Роль такого удостоверения и играет цифровой сертификат. Цифровой сертификат удостоверяет связь между открытым ключом и некоторым объектом. Объектами, с которыми связывается открытый ключ, могут быть имя человека и (или) адрес его электронной почты, доменное имя сайта или название компании-разработчика ПО. Естественно, сертификат, удостоверяющий принадлежность открытого ключа некоей сущности, может быть полезен только при том условии, что он выдан организацией, которой доверяют все участники транзакции. Такие организации именуются в англоязычных странах certificate authorities (CA), а у нас – центрами сертификации или удостоверяющими центрами (УЦ). Для того, чтобы добиться доверия Боба, Алиса должна заверить свой открытый ключ сертификатом, выданным солидной организацией, которой доверяет Боб, и предоставить этот сертификат Бобу для того, чтобы он мог удостовериться в подлинности ее подписи.

Отметим еще раз, что сертификаты не являются неотъемлемой частью систем цифровой подписи и шифрования. Цифровая подпись и шифрование документов выполняются с помощью алгоритмов типа »

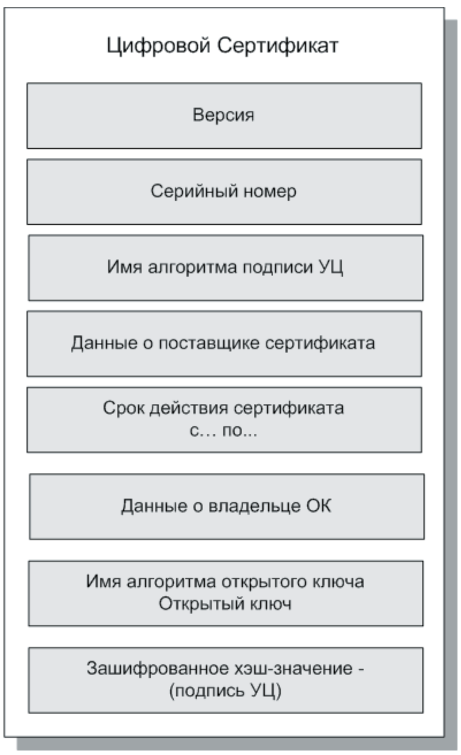
» RSA и их ключей, однако само шифрование (и особенно цифровая подпись) часто имеют смысл только при условии, что участники транзакции доверяют используемым ключам. В этой ситуации цифровые сертификаты оказываются действительно необходимы.

Структура цифрового сертификата

Помимо открытого ключа и имени, с которым ассоциирован ключ, сертификат содержит ряд других важных сведений (**Рис. 1**). Каждый сертификат включает в себя данные о выдавшем его УЦ, необходимые для проверки подлинности. Кроме того, у каждого сертификата есть серийный номер и срок действия (по окончании его сертификат, если он еще нужен, может быть продлен). Если сертификат оказался не нужен (или скомпрометирован) до окончания срока действия, он должен быть отозван в УЦ. Отозванный сертификат теряет юридическую силу.

Подлинность сертификата подтверждается цифровой подписью УЦ. УЦ подписывает свои сертификаты точно так же, как в приведенном выше примере Алиса подписывает сообщения, адресованные Бобу (то есть вычисляет хэш-значение сертификата и шифрует его своим секретным ключом). Секретный ключ крупного УЦ должен охраняться как величайшая тайна, ведь тот, кто получит доступ к этому ключу, сможет подписывать сертификаты от имени УЦ (а значит, получит широчайшие возможности в плане распространения фальшивых электронных подписей).

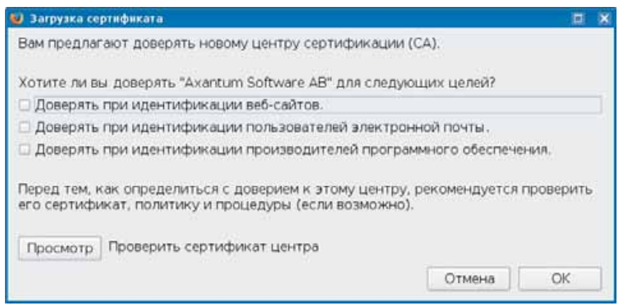
Людей, которые стремятся к абсолютной гарантии, цифровые сертификаты могут довести до отчаяния. Как проверить подлинность подписи УЦ? Для этого нужно расшифровать подпись сертификата с помощью открытого ключа УЦ. А как проверить подлинность открытого ключа УЦ? Публичный ключ УЦ, выдавшего сертификат, может быть заверен сертификатом УЦ более высокого уровня (так образуется цепочка сертификатов, называемая путем сертификации). Но эта цепочка должна где-то заканчиваться! Фактически, мы оказываемся перед той же самой проблемой подлинности открытых ключей, от которой пытались уйти с помощью сертификатов.



› **Рис. 1. Структура сертификата PKI.**

Хотя сертификаты и не гарантируют 100% надежности (а кто нам вообще ее гарантирует в этом Мире?), существует надежная с практической точки зрения система защиты от подделки сертификатов. Речь идет о корневых сертификатах. Корневой сертификат (КС) является корнем пути сертификации и содержит открытый ключ выпустившего его УЦ. Корневой сертификат подписывается с помощью секретного ключа того самого УЦ, чей открытый ключ он удостоверяет (поскольку открытый ключ в корневом сертификате подписан с помощью соответствующего ему секретного ключа, такие сертификаты называют самоподписанными). Фактически КС удостоверяет сам себя, поэтому проверить его достоверность с помощью какого-либо другого сертификата невозможно. Выигрыш от применения КС основан на том, что один КС может использоваться для удостоверения подлинности многих клиентских сертификатов. В результате нам приходится доверять не миллионам сертификатов пользователей, а нескольким десяткам КС (организовать надежное распространение нескольких КС не так уж и трудно).

Каждый раз, когда ваша система удостоверяет пользовательский сертификат, она использует для этого один из КС. Откуда берутся корневые сертификаты в вашей системе? Обычно они поставляются вместе с программным или аппаратным обеспечением (например, в составе дистрибутивов web-браузеров и почтовых клиентов). Это, конечно, означает, что разработчик или поставщик ПО и оборудования могут смошенничать с корневыми сертификатами, но они ведь могут смошенничать и многими другими способами, поэтому если вы вообще доверяете своим программам и устройствам, у вас нет особых причин не доверять включенным в него КС. Хотя ваш браузер и почтовый клиент содержат множество предустановленных КС, время от времени вы можете столкнуться с сертификатом, который удостоверен УЦ, неизвестным вашей системе (иначе говоря, в системе не установлен КС удостоверяющего центра, выдавшего данный сертификат). Встречая сертификат, для которого не предустановлен КС, программы предупреждают вас об этом (**Рис. 2**). Для того, чтобы система смогла проверить подлинность сертификата, вам придется установить удостоверяющий его КС. Важно понимать, что система не станет устанавливать новые КС автоматически (в этом и заключается смысл предупреждающего сообщения, показанного на рисунке). Вы сами должны решить, доверяете ли вы новому удостоверяющему центру и (что еще важнее) источнику, из которого вы собираетесь установить его КС.



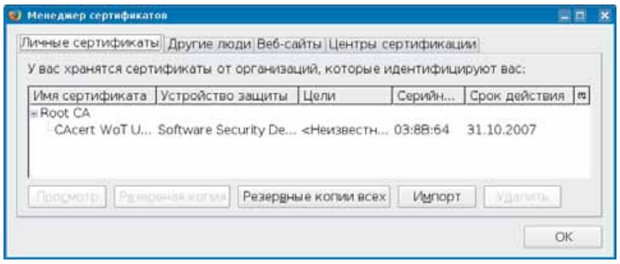
› **Рис. 2. Предложение установить корневой сертификат.**

Таким образом, получив подписанное сообщение и сертификат, удостоверяющий цифровую подпись, адресат сообщения проверяет подпись с помощью полученного сертификата, а сам сертификат – с помощью установленного в его системе КС.

Сертификаты в почтовых клиентах и Web-браузерах

Перейдем от теории к практике. Мы много говорили о сертификатах, пришла пора взглянуть на них собственными глазами. Запустите

браузер *Firefox*. Откройте окно Настройки Firefox и перейдите в группу настроек Дополнительно. Перейдите на вкладку Шифрование и щелкните кнопку Просмотр сертификатов. Откроется окно менеджера сертификатов *Firefox* (**Рис. 3**). Между прочим, менеджер сертификатов *Firefox* позволяет не только просматривать, но и устанавливать, и экспортировать сертификаты. Сертификаты в окне менеджера сгруппированы по несколькими вкладкам. Каждому сертификату соответствует древовидная структура, отражающая путь сертификации, заканчивающийся именем удостоверяющего центра. Обратите внимание, что сертификаты, установленные в *Firefox*, разделены по категориям: личные сертификаты, сертификаты других людей, сертификаты сайтов и сертификаты центров сертификации (корневые сертификаты). Наибольший интерес для нас представляют личные сертификаты (у вас, скорее всего, личных сертификатов еще нет, но скоро вы узнаете, как их получить). Личными считаются сертификаты, установленные вместе с секретным ключом. Поскольку для создания подписи нужен секретный ключ, удостоверить ее можно только личным сертификатом. В вашей системе могут быть установлены не-корневые сертификаты, для которых у вас нет секретного ключа (они располагаются в категориях «сертификаты других людей» или «сертификаты сайтов»). Эти сертификаты можно использовать при шифровании сообщений (с помощью содержащегося в них открытого ключа), предназначенных для владельца секретного ключа.



› Рис. 3. Менеджер сертификатов *Firefox*.

Для передачи сообщений, подписанных и зашифрованных с помощью ключей и сертификатов PKI, используется протокол S/MIME. К подписанному сообщению протокол S/MIME прикрепляет сертификат, удостоверяющий подпись, так что, получив подписанное письмо, вы получаете и открытый ключ его отправителя. Вы можете зашифровать свой ответ, используя этот открытый ключ, и снова передать зашифрованное сообщение по протоколу S/MIME. Тот, кто подписал адресованное вам письмо, сможет расшифровать ваш ответ с помощью своего секретного ключа. Поскольку асимметричные алгоритмы шифрования ресурсоемки, сам документ обычно шифруется с помощью менее требовательного симметричного алгоритма, а открытый ключ применяется только для шифрования «симметричного» ключа.

В заключение этой вводной части рассмотрим фрагмент настоящего сертификата:

```
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 0 (0x0)
Signature Algorithm: md5WithRSAEncryption
Issuer: O=Root CA, OU=http://www.cacert.org, CN=CA Cert Signing
Authority/emailAddress=support@cacert.org
Validity
Not Before: Mar 30 12:29:49 2003 GMT
Not After : Mar 29 12:29:49 2033 GMT
Subject: O=Root CA, OU=http://www.cacert.org, CN=CA Cert Signing
Authority/emailAddress=support@cacert.org
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (4096 bit)
Modulus (4096 bit):
...
```

Этот сертификат является КС удостоверяющего центра CA Cert (обратите внимание, что УЦ выдает этот сертификат самому себе). Как и все КС, этот сертификат обладает длительным сроком действия (до 2033 года). Интересно также отметить, что сертификат был выпущен в 2003 году, то есть тогда, когда об уязвимости алгоритма MD5 еще не было известно, поэтому цифровая подпись сертификата получена с помощью алгоритма MD5, о чем свидетельствует строка

```
Signature Algorithm: md5WithRSAEncryption
```

В принципе это означает, что можно выпустить подложный КС с той же цифровой подписью, однако толку от этого будет немного (подложные КС можно изготовить многими другими способами, но для распространения таких КС нужен надежный канал, которому доверяет множество пользователей). Что же касается не-корневых сертификатов, для которых подлог был бы действительно опасен, то все они сейчас подписываются с помощью SHA1.

На этом мы заканчиваем общее знакомство с сертификатами PKI. В следующей части вы узнаете, как работать с цифровыми сертификатами (и ключами, которые они удостоверяют) в наиболее популярных программах Linux. LXF

» Через месяц Мы закрепим изученную теорию практикой и узнаем, как использовать цифровую подпись в популярных приложениях Linux.

Инструкция

ЧАСТЬ 2 Устанавливайте поудобнее свою камеру, включайте проектор – **Петр Семилетов** готов научить вас паре изящных приемов

Сегодня мы познакомимся с некоторыми важными функциями *Cinelerra*, которые не раз пригодятся нам в дальнейшем. Это своеобразная азбука, без которой дальше двигаться нельзя.

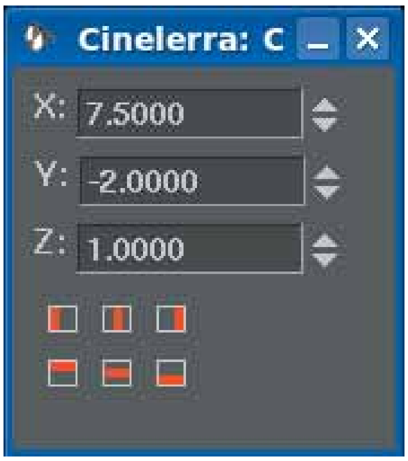
Известно, что *Cinelerra* оснащена функциями не только монтирования (т.е. расположения кадров на монтажном столе), но и составления, или, как сейчас говорят, «композитинга». Последнее обстоятельство приближает *Cinelerra* к таким продуктам, как *Shaker* или *Adobe After Effects*. То, что в *Cinelerra* обе составляющие – монтирования и композитинга – соединены, дает нам большую гибкость в работе.

Камера и проектор

Функции окна составителя (Compositor) не ограничиваются выводом в него видео – в нем есть две такие штуки, как камера и проектор. Инструмент управления камерой – четвертый сверху в окне Compositor, а инструмент проектора – следующий за ним, то есть пятый. Кроме того, настраивать камеру и проектор можно с помощью кривых на монтажном столе. Чтобы включить кривые управления камерой, перейдите в меню View и поставьте галочки на Camera X, Camera Y и Camera Z. Каждая кривая соответствует оси координат. Так, поставив точку (она называется ключевым кадром) на кривой Camera X и перемещая эту точку, мы сдвигаем объектив камеры по горизонтали.

Чтобы перемещать камеру с точностью до пиксела, а также использовать выравнивание камеры относительно заданной оси, можно воспользоваться окном свойств инструмента. Оно вызывается кнопкой Show Info Tool (вторая снизу на той же панели, выглядит как знак вопроса).

> **Рис. 1. Параметры камеры.**



Виртуальной камерой мы указываем составителю область, которую надо взять из исходного изображения и вывести через виртуальный проектор в конечный кадр, то есть в то, что мы видим в окне составителя.

Для чего нужна виртуальная камера? Ее применение очень разнообразно. Допустим, у вас есть фоновое изображение, большее по размеру, чем выходной кадр. Смещая камеру по горизонтали, мы можем плавно панорамировать изображение – вспомните концовку первой части DOOM. Или, например, где-то в другой программе были нарисованы (или нарисованы и отсканированы) красивые титры со всякими изысками, причем выполнены они в виде эдакой вертикальной ленты. Двигая камеру по вертикали, их можно анимировать и заставить перемещаться снизу вверх. А сдвигая камеру по оси Z, можно приблизить или отодвинуть изображение – это хороший способ масштабирования.

Обратите внимание, что смещение изображения относительно виртуального видоискателя всегда противоположно движению камеры. То есть – смещаем камеру влево, а картинка перемещается вправо.

Что такое проектор? Его задача – указать, в какую область конечного кадра надо вывести изображение. Кривые проектора находим в том же меню View -> Projector X, Projector Y и Projector Z. Но если смещение камеры – это перемещение окна обзора внутри картинки, то смещение проектора – это перемещение самой картинки. Именно проектор мы и используем, когда хотим поместить объект (видеокадр или статичное изображение) в нужное нам место на экране.

И камера, и проектор поддерживают автоматизацию –динамическое изменение своих параметров. Для этого служат ключевые кадры (keyframes). Они только называются «кадрами», а на самом деле это просто точки на кривых, отображаемых прямо на дорожках в окне монтажного стола. В каждой такой точке хранится значение параметра в данной временной позиции. Например, сделаем на кривой Camera Z две точки (два ключевых кадра) – один на энной секунде видео, а другой – секунды на полторы дальше. Первую точку оставим без изменений, а вторую переместим куда-то вверх (что равнозначно приближению, наплыву). Теперь при воспроизведении видео в этом отрезке камера начнет показывать его вначале нормально, а затем в течение полутора секунд плавно, но быстро приблизит его.

Работа с масками

В окне составителя можно использовать очень мощное средство, именуемое маской. Маска – это, грубо говоря, область, которая ограничивает вывод текущей дорожки в проектор. В зависимости от типа маски, можно направлять в проектор изображение внутри маски, либо ту часть кадра, которая снаружи маски, вне ее. По умолчанию все, что вне маски, становится прозрачным.

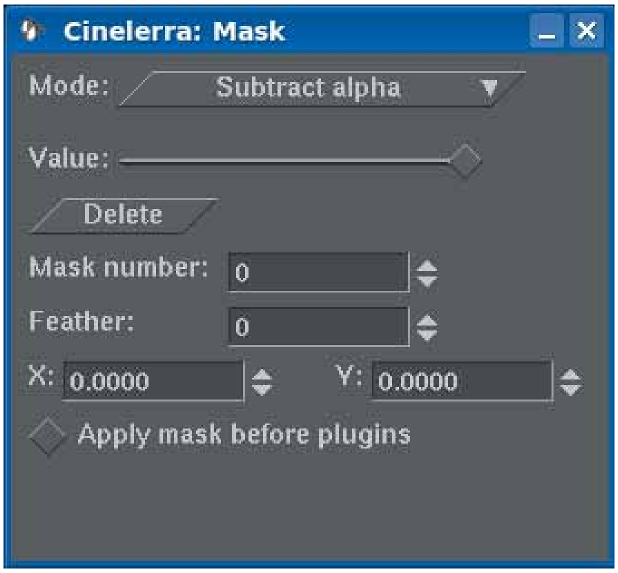
Чтобы создать маску, надо выбрать инструмент Edit Mask на панели инструментов в окне Compositor. Кнопка этого инструмента – третья сверху. Указатель мыши перейдет в режим редактирования маски. Граница маски состоит из узлов. Вы можете создавать новые узлы и перемещать существующие. Удерживая Shift и перетаскивая узлы, вы сделаете границы маски эллиптическими, а без Shift'а – прямыми. Чтобы переместить маску, надо, удерживая Alt и Ctrl, взять ее за один из узлов и сдвинуть в нужное положение. А удерживая Ctrl, можно перемещать узлы.

Месяц назад Мы использовали хроматический ключ, чтобы открыть дверь, ведущую к тайнам кинематографа.

к применению

На дорожке может быть несколько масок, но не более восьми.

Для управления параметрами маски надо нажать кнопку Show Info Tool. Давайте изучим открывшееся окно подробнее:



› **Рис. 2.**

» **Mask number** Устанавливает номер текущей маски. По умолчанию маска только одна (номер ноль). Увеличивая счетчик масок, можно создавать дополнительные маски либо переключать уже созданные. Все маски одной дорожки разделяют между собой параметры. Зачем может потребоваться несколько масок? Чтобы было удобнее придать нужную форму ограничиваемой области.

» **X, Y** Здесь задаются координаты текущего узла маски – с точностью до тысячных долей пиксела. Это средство используется, если вы по какой-то причине не можете точно работать с маской при помощи мыши.

» **Mode** Режим маски, доступно два значения – Multiply alpha и Subtract alpha. В первом случае в проектор направляется часть кадра, находящаяся внутри маски. А при Subtract alpha в проектор попадает то, что снаружи маски, а сама маска становится прозрачной «дыркой».

» **Value** Уровень прозрачности изображения в маске или вне ее.

» **Feather** Степень размытия границы маски.

Теперь поговорим о практическом использовании масок. Вот пример – допустим, вы хотите наложить эффект на некоторую область кадра, а не на весь кадр. Что делать?

На ум сразу приходит GIMP, где мы выделяем область и применяем эффект к ней. Но ведь маска, по сути, то же выделение! Однако есть подводный камень. Изображение внутри маски будет обработано эффектом, но все, что лежит вне маски, станет прозрачным. Обходится это очень просто:

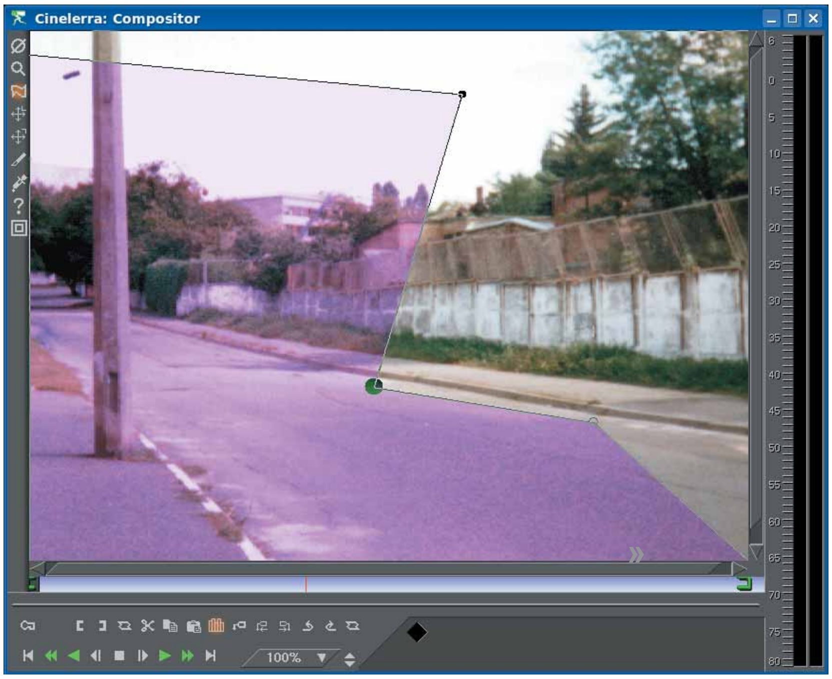
1 Создайте, одну над другой, две дорожки с одинаковым содержимым. Важно, чтобы параметры камеры и проектора этих дорожек тоже были одинаковыми (задайте их точно в Info Tool).

2 В нижней дорожке отключите кнопку Arm, чтобы сделать активной верхнюю дорожку.

3 Нарисуйте маску на верхней дорожке. Что получилось? Пока ничего осязаемого. Хотя изображение вне маски стало прозрачным, мы не видим этого, поскольку под верхней дорожкой лежит точно такая же нижняя дорожка.

4 Наложите эффект на нижнюю дорожку. Вот тут-то и проявилась маска!

5 В итоге получается обработанный в маске участок кадра, а вокруг – нетронутое изображение с кадра-подложки.



› **Рис. 3. К участку кадра применен эффект «Цветовой баланс».**

Как это можно использовать на практике? Вы наверняка видели в телепередачах вроде разных криминальных хроник, как при показе свидетелей преступления или самих преступников им пикселизируют лица (либо применяют размытие). Также изменяют голос – в простейшем случае методом изменения питча – базовой частоты. В *Cinelerra* есть такой звуковой эффект, называется он просто – Pitch shift. С его помощью голос можно сделать выше или ниже. Но это обратимо, и при желании оригинальный голос можно восстановить. Чтобы исказить голос так, чтобы его вообще никто не узнал, надо использовать

вокодер, которого в *Cinelerra* нет. Но вернемся к видео. Давайте пикселизируем лицо в кадре. Вот алгоритм:

1 Уже привычно создаем две дорожки с одинаковым содержимым, отключаем Arm Track для нижней дорожки.

2 Клавишами [и] отмечаем на дорожке начало и конец обрабатываемого фрагмента.

3 Ставим курсор в начало фрагмента, где будет пикселизироваться лицо героя.

4 В составителе рисуем на лице героя маску и устанавливаем для нее режим Multiply alpha.

5 Применяем к верхней дорожке эффект Downsample. Он принимает ряд параметров. Передвигая их ползунки, добиваемся нужной степени пикселизации.

6 Теперь обсудим тонкость – человек может перемещаться в кадре. Как маске уследить за ним? Поможет автоматизация – динамическое изменение параметров во времени. В окне монтажного стола или в окне Compositor включаем кнопку с изображением ключика.

7 Перемещая по дорожке курсор (тем самым изменяя текущий кадр), двигаем маску в окне Compositor мышью, зажав клавиши Alt и Ctrl. Все перемещения будут сохранены и повторены при воспроизведении.

Это лишь один практический пример использования масок. Уверен, вы изобретете десятки других, оптимизированных под конкретные нужды. Главное, что *Cinelerra* предоставляет в ваше пользование такой мощный инструмент. Ведь в том же *Adobe Premiere Pro 2.x* я не могу выделить произвольный участок кадра нужной мне формы. Для этого мне пришлось бы задействовать *Adobe After Effects*. А в *Cinelerra* – пожалуйста, все в одном флаконе.

Режимы наложения дорожек

В окне монтажного стола, слева от каждой дорожки находятся ее свойства или дополнительные параметры (если их не видно, нажмите кнопку с изображение треугольной стрелки, она крайняя справа). Вы увидите текстовое поле Nudge (в нем можно задавать задержку/опережение видео или звука при воспроизведении) и пиктограмму Overlay (выглядит как ряд прямоугольников). Overlay означает «наложение». Под пиктограммой скрывается список, в котором доступны режимы наложения.

То, как дорожка накладывается на лежащую под ней дорожку, зависит от содержимого этих дорожек. Назовем их дорожка А (верхняя) и дорожка Б (нижняя). Вот некоторые результаты наложения:

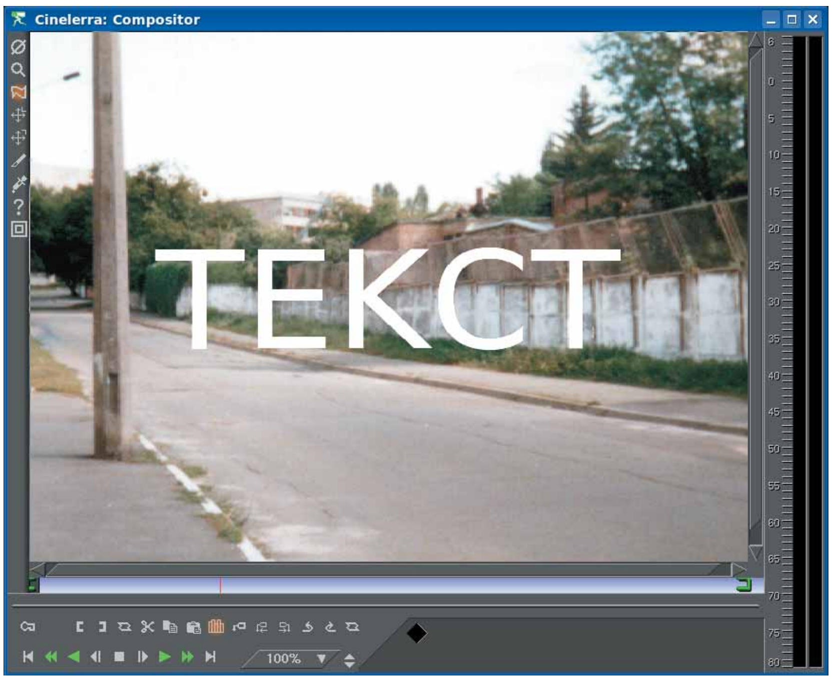
1 Если на дорожке А помещено изображение с альфа-каналом, а режим наложения равен Normal, то альфа-область становится прозрачной.

2 Если на дорожке А расположен объект на черном фоне, и выбран режим Addition, то черный фон станет прозрачным. Иногда это полезно, если вы сделали титры (белого цвета) в программе, которая не может выводить результат с альфа-каналом, и пришлось сохранять титры на черном фоне.

3 И напротив, если при таком же раскладе выбрать режим Multiply, то черный фон останется, а отличные от него объекты (т.е. буквы) станут прозрачными. Цвета объекта, отличные от белого, будут так или иначе искажены.

Замечу, что в документации с *Cinelerra* сведения о режимах наложения ошибочны – опыт дает результаты, отличные от описанных в руководстве. Так, в нем сказано, что с черно-белой картинкой режим Multiply убирает черный фон, оставляя только белый. По моим наблюдениям, такой эффект достигается режимом Addition, а, как сказано выше, режим Multiply влияет на картинку иначе – делая белый цвет прозрачным и оставляя черный. Для сравнения можно взять *Adobe After Effects*, там у дорожки в контекстном меню есть список Blendind – аналог Overlay, а режимы Addition и Multiply работают описанным выше способом.

Режимы наложения ведут себя предсказуемо при работе с черно-белыми изображениями, которые вы хотите наложить на цветную подложку. В противном случае перед нами открываются широкие возможности по созданию различных цветовых эффектов. LXF



› Рис. 4. Этот текст, нарисованный на черном фоне, выведен в режиме Addition, чтобы фон стал прозрачным

Через месяц Все любят мультики – и мы поговорим о создании и монтаже анимации.

MEncoder: ужимаем видео

Цифровой видеопоток может быть сжат различными алгоритмами или не сжат вовсе. Всё это влияет как на размеры видеофайла, так и на качество изображения. Найти золотую середину между двумя крайностями поможет **Евгений Гаврилов**.



Наш эксперт

Евгений Гаврилов
Евгений использует Linux для решения самых разных задач и верен, что консоль куда гибче и сильнее, чем любой GUI.

У некоторых читателей, вероятно, уже возник вопрос: «А зачем кодировать видео? Ведь это потеря качества!». Ну, во-первых, потеря качества бывает и незначительной, а при использовании фильтров возможно даже некоторое улучшение изображения, зато размер файла сокращается в разы. К тому же, стандартные сжатые AVI-файлы гораздо проще передавать через Интернет. Вы же не хотите, чтобы ваши родственники и друзья умерли в ожидании загрузки DVD-диска объемом 4,5 ГБ с записью вашего новорожденного ребенка? Тогда читайте дальше!

Форматы и кодеки

На сегодняшний день существует множество кодеков (т.е. способов сжать видеопоток), и форматов (средств упаковки сжатых данных в файл вместе с сопутствующей информацией). Вездесущая корпорация Microsoft призывает к использованию формата WMV (Windows Media Video), так как он (по ее мнению) значительно лучше некогда ею же созданного контейнера AVI *1. Многочисленные пользователи, тем не менее, пока не собираются переходить на новый формат. Мы же не будем пользоваться и AVI, а обратим свое внимание на OGM (*Ogg Media*, или *Ogg Media File*). Он позволяет без проблем хранить звук, сжатый кодеком *Vorbis*.

Отметим, что *Ogg* – это контейнер, а *Vorbis* – алгоритм сжатия. Говоря «*Ogg Vorbis*», обычно подразумевают файл-контейнер *Ogg*, звук в котором сжат при помощи *Vorbis*. Далее, для краткости, мы будем использовать «*Ogg*» вместо «*Ogg Vorbis*». Что же касается OGM, то он возник как развитие контейнера *Ogg* и позволяет хранить данные, сжатые видео- и звуковыми кодеками, которых на сегодняшний день существует не меньше, чем различных форматов контейнеров. Наиболее популярными являются: *Xvid* и DivX для видео, *Vorbis* и MP3 – для аудио. Все эти форматы сегодня в равной степени поддерживаются аппаратно (в бытовой технике), кроме разве что *Ogg Vorbis*, который покамест понимается весьма ограниченным набором «реальных» проигрывателей. Кроме того, редкий аппаратный плейер умеет работать с контейнером OGM.

Мы остановим свой выбор на *Xvid* и *Vorbis*. Причин этому несколько. Во-первых, оба кодека распространяются по лицензии GPL. Во-вторых, если *Xvid* и DivX еще могут посоревноваться за пальму первенства по качеству получаемого видео, то со звуком ситуация проще. *Vorbis* разрабатывался несколько позже MP3, что дало ему определенные преимущества: разработчики смогли учесть те моменты, которые не были приняты во внимание при создании MP3. К тому же, *Vorbis* использует другую психоакустическую модель, обеспечивающую более высокое качество.

Кодировщики

Для работы с видеофайлами нам понадобится *MPlayer*, в состав которого входит отличный кодировщик *mencoder*. Для него существует множество графических оболочек, но использовать их я бы не советовал: удобный интерфейс редко включает все доступные опции, а в результате вы теряете контроль над процессом.

Перед установкой самого *MPlayer* необходимо проинсталлировать кодеки *Xvid* и *Vorbis* – обычно они находятся в пакете essentail-codecs. Не забудьте добавить путь к ним в файл **/etc/ld.so.conf** и указать их каталог опцией --with-codecsdir=DIR при сборке *MPlayer* (если вы хотите получить стандартный графический интерфейс к *MPlayer*, не забудьте также ключ --enable-gui). После выполнения configure, внимательно посмотрите отчет о кодеках, поддерживаемых *MPlayer* в вашей системе.

Что кодировать?

Перекодирование можно осуществлять между любыми двумя форматами, при условии, что они поддерживаются *MPlayer*. Однако, в большинстве случаев исходным форматом является MPEG-2 (этот способ сжатия используется в DVD), а выходным – MPEG-4 (например, *Xvid*). Сформулируем стоящую перед нами задачу следующим образом: необходимо перекодировать DVD (видеозапись хранится в файлах **video1.vob**, **video2.vob**) в AVI. Вам все понятно? Тогда приступим к решению!

Вначале для удобства работы объединим два VOB-файла в один:

```
$ cat video1.vob video2.vob|mencoder - -oac copy -ovc copy -o video.vob -ffourcc MPEG-2
```

Мы «сливаем» **video[1-2].vob** простой cat и затем копируем их аудио- (-oac copy) и видеопотоки (-ovc copy) с помощью *mencoder*. Для того, чтобы полученный файл можно было просмотреть не только в *MPlayer*, рекомендуется сделать запись об используемом кодеке в заголовке файла: для этого применяется опция -ffourcc. Но поскольку наш файл **video.vob** является промежуточным, это не так уж важно. На данном этапе из фильма можно также «вырезать» определенные моменты, воспользовавшись опциями -ss <первый кадр> и -endpos <продолжительность или размер>. Продолжительность задается в стандартном формате [[hh:]mm:]ss[.ms], а размер – в байтах (b), килобайтах (kb) или мегабайтах (mb).

Шаг 1: Звук

Теперь, когда мы получили объемный, но единый файл, можно приступать к первой фазе – извлечению аудиодорожки и сжатию ее в *Ogg*. Преобразование аудиопотока из одного сжатого формата в другой напрямую невозможно, хотя некоторые программы и утверждают, что умеют делать это. На самом деле, «за кулисами» звук преобразуется в несжатый формат PCM – Pulse-coding modulation – а затем обрабатывается выходным кодеком. Мы пойдем тем же путем. Чтобы извлечь PCM-аудиопоток из файла **video.vob**, можно выполнить следующую команду:

```
$ mplayer video.vob -ao pcm:fast:file=track.wav -vo null
```

Фактически, мы воспроизводим полученный файл в *MPlayer*, однако, для вывода звука и изображения используются не совсем стандартные устройства. Звуковой драйвер указывается опцией -ao. Здесь мы выбрали PCM с сохранением в файл **track.wav** (вместо отправки на привычные колонки), а опция fast позволила нам сделать это «быстро» (т.е. не в режиме реального времени воспроизведения фильма). Параметр -vo null задает нуль-устройство для вывода изображения, т.е. отсекает ненужное на данном этапе видео.

Полученный звук можно закодировать с помощью утилиты *Oggenc*, входящей в пакет *Vorbis*.

```
#oggenc track.wav -q 3 -o track.ogg
```

Параметр -q 3 задает уровень качества. Номинальный битрейт составляет 112 кбит/с, средний – 88 кбит/с. Не пугайтесь низких значений – вспомните, что *Vorbis* использует психоакустическую модель, отличную от применяемой в MP3, поэтому «эффективный» битрейт составляет около 128 кбит/с. При всем этом *Ogg* занимает меньше места, так что при желании можно поднять качество и до 4.

Закодированный звук сохраняется в файле **track.ogg**. Прослушайте его, и если результат вас устраивает, освободите место на диске, удалив файл **track.wav**.

Шаг 2: Видео

Теперь приступим к обработке, а точнее – к кодированию видео с помощью *mencoder*. Одним из главных параметров здесь является все тот же битрейт. Если вам неважен размер конечного файла, то все определяется требуемым качеством: при битрейте 800 кбит/с картинка выглядит вполне приемлемо. Если же нужно получить файл определенных размеров, то следует воспользоваться специальным калькулятором, встроенным в различные графические оболочки. Впрочем, все вычисления можно произвести прямо в командной строке – главное знать математику на уровне 6 класса и *Bash* (или что там вы выбрали, прочитав наше **Сравнение**?) на уровне продвинутого пользователя. Настоящие эстеты могут воспользоваться *bc*. В общем, формула такова:

```
Mb=(length_min*60*bps/8)/1024
```

или, в Bash-нотации:

```
echo «$[($length_min*60*$bps/8)/1024] Mb required»
```

Мы переводим длину фильма (в минутах) в секунды и умножаем на битрейт, деленный на 8, поскольку он измеряется в килобитах. Поделив результат на 1024, имеем количество мегабайт, которое будет занимать наш фильм. Заметьте, мы говорим только о видео. Не забывайте добавить к полученному числу размер аудиопотока.

Из данной формулы легко получить еще одну – для подсчета битрейта, обеспечивающего требуемый размер ролика:

```
bps=(Mb*1024)/length_min/60*8
```

Полезно оценить порядки этих величин. Допустим, фильм идет 90 минут и мы хотим записать его на стандартный CD-R емкостью 700 МБ. Размер звуковой дорожки – 100 МБ. Набрав в консоли:

```
#echo $[600*1024/90/60*8]
```

получаем ответ: 904 кбит/с. Теперь можно приступать к кодированию:

```
$ mencoder video.vob -oac copy -ovc xvid -xvidencopts bitrate=904 -o
video.avi -ffourcc xvid -forceidx
```

Требуемый кодек (*Xvid*) указывается опцией -ovc (option video codec), а параметр -xvidencopts определяет его настройки. В данном случае мы выбираем битрейт 904 кбит/с. Звук покамест оставим без изменения – *mencoder* с трудом справляется с мультиплексированием (т.е. сведением нескольких потоков данных в один) видео и звука *Vorbis* в контейнер AVI.

На данном этапе можно также применить один или несколько фильтров: обрезать поля, изменить соотношение сторон, выполнить деинтерлейсинг и т.п. Полный список фильтров, поддерживаемых *mencoder*, выводится командой

```
# mencoder -vf help
```

а подробную информацию о каком-то фильтре дает команда

```
# mencoder -vf <filter>=help
```

Черные поля вокруг кадра лучше удалить сразу. Они неплохо сжимаются (в смысле размера файла), но все же они занимают драгоценное место на носителе – да и выглядят неопрятно. Для этих целей служит фильтр crop, который принимает в качестве параметров ширину и высоту конечной картинки. Например, в предыдущую команду можно добавить:

```
# mencoder ... -o video.vob -vf crop=640:400 ...
```

Троеточие обозначает опущенные (совпадающие) опции. Вы также можете переложить всю «черную работу» на фильтр автоокадрирования cropdetect:

```
# mencoder ... -o video.vob -vf cropdetect ...
```

Если же вы хотите отмасштабировать изображение (уменьшить или увеличить размер картинки), на помощь придет фильтр scale. Его вызов таков:

```
-vf scale=w:h
```

Здесь w и h – новые ширина и высота изображения. Имейте в виду, что фильтры передают обрабатываемое изображение «по цепочке», и порядок их следования в ней имеет значение.

Чтобы после кодирования изображение не было приплюснуто или растянуто и наши люди не получились яйцеголовыми, следует воспользоваться опцией -xvidencopts aspect=<aspect>. В качестве параметра <aspect> можно указать соотношение сторон в виде дроби (4/3) или числа с плавающей точкой. Для его автоматического определения используется параметр autoaspect, а для указания соотношения в контейнере AVI – опция mencoder --force-avi-aspect <0.2-3.0>.

Рассмотрим изученные выше фильтры и опции на примере. Допустим, размер изображения составляет 640x480, а черные «поля» имеют ширину 20 пикселей. Мы обрежем их, растянем изображение до 720 пикселей в ширину и установим соотношение сторон равным 16/9. С учетом всех этих требований, командная строка будет выглядеть следующим образом:

```
# mencoder video.vob -vf crop=600:480,scale=720 -oac copy -ovc xvid -
xvidencopts bitrate=904:aspect=16/9 -o video.avi -ffourcc xvid -forceidx
```

Пользоваться -xvidencopts autoaspect после масштабирования и кадрирования бессмысленно – программа, скорее всего, ошибется.

Скорая помощь

Чтобы просто оценить результаты работы *mencoder*, не обязательно дожидаться окончания кодирования. Нажмите Ctrl+C и вы сможете быстро просмотреть получившийся отрывок.

Шаг 3: Мультиплексирование

Итак, мы имеем звуковую дорожку в отдельном файле *Ogg Vorbis*, а также обработанное должным образом видео в контейнере AVI. Остался последний этап – мультиплексирование. Как мы и договаривались в начале статьи, сводить звук и видео мы будем в контейнере OGM. Для этих целей мы воспользуемся пакетом *ogmtools*, а именно, утилитой *ogmmerge*, назначение которой вполне понятно из ее названия. В качестве параметров *ogmmerge* принимает имена входных и выходного файлов.

```
# ogmmerge -o final.ogm -A video.avi track.ogg
```

После объединения двух потоков будет получен OGM-файл, который можно без проблем воспроизводить практически в любой программе. Опция -A отбрасывает аудиопоток, сохраненный в контейнере **video.avi**. Теперь самое время опубликовать файл **final.ogm** в Сети или его записать на чистый компакт-диск и раздать друзьям. Если вам удастся заснять что-то стоящее, не забудьте поделиться ссылочкой и с нами (letters@linuxformat.ru)! LXF

Альтернатива: FFmpeg

FFmpeg – это законченное решение для записи, преобразования и воспроизведения аудио- и видеоконтента. В его основе лежит библиотека мультимедиа-кодеков, известная как *libavcodec*. Большинство из входящих в нее кодеков было разработано с нуля, чтобы обеспечить лучшую производительность.

Для того, чтобы использовать какой-либо MPEG4-совместимый кодек из числа предлагаемых *libavcodec* совместно с mencoder, необходимо указать соответствующее семейство опцией **-ovc lavc** и задать параметры кодирования ключом **-lavcopts**:

```
# mencoder VTS01.VOB -o home.avi -ovc lavc -lavcopts vcodec=xvid:vbitrate=900
```

Здесь мы выбрали кодек *Xvid* и установили битрейт равным 900 кбит/сек. Остальные подробности можно найти по адресу: http://ffmpeg.mplayerhq.hu/.

Альтернатива: x264

x264 – это свободная библиотека для кодирования видеопотоков H264/AVC. Для того, чтобы *mencoder* мог использовать *x264*, его, скорее всего, придется пересобрать. Конечно, предварительно нужно установить сам x264. Контейнеры AVI и OGG/OGM не могут вмещать в себя потоки H264/AVC, поэтому для них лучше использовать **mp4** (Mpeg4) либо **mkv** (Matroska).

Для мультиплексирования выходного файла можно воспользоваться услугами библиотеки *libavformat* (*-of lavf*), определяющей формат файла по расширению, или же задать его вручную опцией format:

```
# mencoder VTS01.vob -o home.mp4 -of lavf -ovc x264
```

В мире IT всегда приходится выбирать между памятью (качеством) и временем выполнения. Кодек *x264* обеспечивает качественное изображение и высокую степень сжатия, но работает очень медленно. Имейте это в виду, если только у вас не современный двухядерный процессор.



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

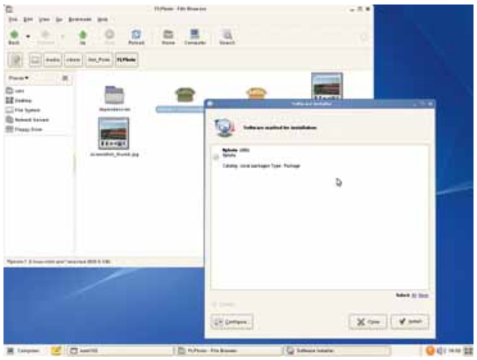
Сегодня мы ответим на вопросы о:

1 Пакетах
2 DHCP
3 Синхронизации почтовых ящиков
4 Switchdesk
5 Инсталляции из исходных текстов
6 Звездочке
7 Модемах
8 Qemu
9 MySQL
10 Почте
11 Белых списках SpamAssassin
12 Восстановлению PPP-соединения
13 Переходе на беспроводную сеть
★ SSH
★ Авторазмонтированию съемных устройств

1 Инсталляция с DVD

В Я полный новичок в Linux, но сумел установить OpenSUSE 10.2 с DVD LXF89. Это нечто! До сих пор под впечатлением от увиденного. Возможно, мой вопрос выдаст мое невежество: как перенести другие программы с DVD (например, *FLPhoto*) на мой ПК, чтобы они работали? Инструкций по перемещению, установке, встраиванию и другим методам сделать программу «инсталлированной» и работающей я не нашел, наверняка потому, что это известно всем, кто пользуется DVD и Linux. Дайте же и мне ниточку.

Михаэль Мюллер [Michael Mueller]



В SUSE и ряде других дистрибутивов RPM с DVD можно установить просто двойным щелчком.

О Инсталлировать программу можно по-разному, в зависимости от способа ее упаковки. В Вашем случае существуют два файла, содержащих *FLPhoto*: **FLPhoto-1.3-source.tar.gz** и **FLPhoto-1.3-linux-intel.rpm**. SUSE использует формат пакетов RPM, поэтому Вам нужен второй файл. Инсталлировать его можно с помощью *Yast* – инструмента администрирования SUSE на все случаи жизни – двойным щелчком на файле. У Вас спросят пароль администратора (установка программы предполагает запись в системные каталоги), затем появится инсталлятор, и останется лишь нажать кнопку «Install» (Установить).

Если RPM-файла нет, то остается возможность установить программу из архива с исходными текстами (эти файлы обычно оканчиваются на **.tar.gz** или **.tar.bz2**) – см. далее. Такая инсталляция предусматривает наличие компилятора, который SUSE по умолчанию не устанавливает. Но он все же присутствует на DVD: все, что Вам нужно – запустить *Yast*, выбрать пункт Software Management (Управление программами), набрать в поле поиска **gcc**, выбрать пакет *gcc* и нажать кнопку «Install». Вместе с пакетом *gcc* будут установлены все компоненты, необходимые для сборки программ из исходных кодов. **НБ**

2 Оптимизация Optus

В Я из Австралии, у меня широкополосное подключение к Интернету через Optus. Не мог заставить Linux подключиться к модему (Siemens SpeedStream 4200), пока не поставил Gentoo 2006.1. Когда все заработало, я заметил такую разницу: Gentoo использует *dhcpcd*, а другие работают через pump либо *dhclient*. Mandriva 2007 предлагает все три, но работает лишь *dhcpcd*. Начав разбираться, я увидел, что *dhcpcd* применяет параметр –h hostname.

Добиться работы других программ я не смог, зато заработали другие дистрибутивы (DSL-N, Knoppix и Ubuntu): я монтировал на них раздел Gentoo и запускал *dhcpcd*, который тут же выдавал сеть. Другие программы получали вроде бы корректный IP-адрес, но не могли подключиться, либо обрывали соединение, прежде чем я мог им воспользоваться. Я думал, что Optus нужен особый модем, но он работает и с другими марками модемов. До того, как я нашел Gentoo, приходилось пользоваться фирменным установочным диском в Windows XP. В чем различие этих программ?

Питер Соренсен [Peter Sorensen]

О Похоже, эта проблема характерна для данного модема. При работе с некоторыми DHCP-программами он ведет себя в точности как Вы описали: получает IP-адрес, а затем обрывает подключение. Решений здесь два, и одно из них Вы уже нашли: пользуясь *dhcpcd*, можно передать модему hostname с параметром **-h**. Эквивалент для pump – это **-u** или **--hostname**, а вот для *dhclient* нет ничего. Может оказаться, что Ваш провайдер или Ваш модем чересчур придирчивы к формату направляемых Вами DHCP-запросов.

Поскольку поддержка DHCP этим модемом откровенно неустойчива, более надежным может оказаться другое решение: постоянный IP-адрес. Необходимо узнать IP-адрес, назначаемый вашему модему – это можно сделать, подключившись через *dhcpcd*, из-под Windows или путем проб и ошибок. Конкретный адрес зависит от провайдера, но по умолчанию для OptusNet »

Наши эксперты

» Мы найдем эксперта для ответа на любой вопрос, от инсталляции и установки модемов до администрирования сети – лишь черкните письмо, а об остальном мы позаботимся.



Нейл Ботвик
Поруководив небольшим интернет-провайдером и поделав сопроводительные диски для LXF, Нейл считает себя «Linux-мастером на все руки».



Майк Сондерс
Майк стоял у истоков Linux Answers в LXF. Его специализация – программирование, оконные менеджеры, init-скрипты и SNES.



Пол Хадсон
Пол – местный супер-программист. Он управится со всеми вашими проблемами по части web и баз данных.



Ник Вейч
Когда не редактирует что-либо большим красным карандашом, Ник успевает разъяснить читателям все, что они хотят знать о графическом и 3D ПО для Linux. Готов ответить даже на самый простой вопрос!



Владимир Иванов
Работает с Linux уже более 10 лет, днем – создавая комплексные решения для крупных компаний, а вечером – модерируя сайт linux.org.ru и отвечая на вопросы о системном администрировании и информационной безопасности.



Валентин Синицын
В свободное от работы время редактор нашего журнала разрабатывает *KNetworkManager* и другие открытые приложения. Он с радостью поможет вам в вопросах использования Linux на рабочем столе.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите на м по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

» обычно 10.1.1.1. Узнав IP-адрес, несложно настроить Ethernet-интерфейс Вашего компьютера на использование статического адреса. Возьмите адрес одной подсети с модем (скажем, 10.1.1.2) и назначьте шлюзу и DNS-серверу адрес модема (10.1.1.1). Маску подсети необходимо установить в 255.255.255.0.

С такой настройкой проблемы должны исчезнуть. DHCP здорово сберегает время при работе в крупных сетях или при перемещении от одной сети к другой с ноутбуком. А в небольшой домашней сети лучше выделить каждому устройству постоянный адрес. **НБ**

3 Почта отовсюду

В Использую *Thunderbird* в качестве почтового клиента, и вполне им доволен. Я прикупил два ноутбука и хотел бы иметь доступ к своему почтовому ящику со всех своих машин – например, чтобы пользоваться общим содержимым. Вот что мне нужно, в двух словах:

» Хочу забирать почту с сервера ISP разом (оставлять ее там ни к чему).

» На всех машинах я хочу использовать *Thunderbird* в качестве почтового клиента.

» Все компьютеры должны иметь доступ ко всем моим почтовым ящикам, чтобы, например, отправив сообщение с одного из ноутбуков, я мог видеть отправленное сообщение на втором ноутбуке и на настольном ПК.

» Все это должно работать как в Linux, так и в Mac OS X 10.4.

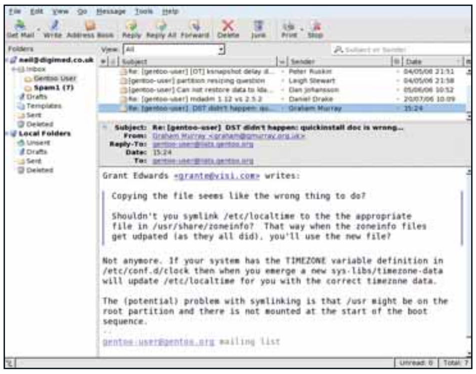
» Мне нужно открытое ПО.

Пробовал просто открыть общий доступ к почтовой директории через *Samba*, но – не сработало, файлы индекса оказались испорчены.

sfinnie, из форумов LXF

О Вы можете добиться своего двумя способами. Один из них – забирать почту с сервера через POP3 и синхронизировать почтовые директории на двух машинах. *Unison* (www.cis.upenn.edu/~bcpierce/Unison) превосходно справляется с такими задачами, а также с синхронизацией любой части Вашей домашней директории, которую Вам хотелось бы обновлять на нескольких компьютерах. *Unison* – лучший выбор для непрерывной синхронизации двух компьютеров: с его помощью мой ноутбук идет в ногу с настольным ПК. *Unison* использует протокол rsync для экономии трафика и времени, но, в отличие от rsync, обрабатывает еще и ситуации, когда после последней синхронизации файлы обновились на каждом из компьютеров. Настройка работы с тремя машинами потребует, вероятно, некоторых дополнительных усилий, но определенно возможна.

Другой вариант, относящийся лишь к почте – это установка собственного IMAP-сервера на Вашем настольном ПК. Для доставки сообщений с сервера провайдера и их локального хранения можно воспользоваться *Fetchmail*, затем указать почтовым программам на каждом компьютере на IMAP-сервер на Вашем настольном ПК (на самом ПК следует проделать то же самое, присвоив серверу localhost). Ваша почта хранится на сервере вместе с информацией о статусе, т.е. когда Вы прочтете сообщение на одном компьютере, оно получит отметку о прочтении на всех остальных. В отличие от POP3, IMAP позволяет хранить почту на сервере и читать ее отовсюду через Интернет. Большинство почтовых клиентов могут синхронизировать локальное хранилище с сервером, поэтому можно будет получить локальные копии сообщений для чтения оффлайн.



› Thunderburd может забирать почту с вашего собственного IMAP-сервера, позволяя делать это с разных ПК.

Я предпочитаю *Dovecot* (http://dovecot.org), хотя лучше, вероятно, воспользоваться сервером, устанавливаемом в Вашем дистрибутиве по умолчанию: большая часть работы по настройке уже проделана во время инсталляции, к тому же вам помогут форумы и списки рассылки дистрибутива. Чтобы выполнить Ваши несложные требования, не нужно особо отклоняться (если это вообще понадобится) от стандартной настройки. *Dovecot* wiki, по приведенному адресу, содержит множество полезной информации. Еще одно преимущество такой системы – если в Вашей семье несколько пользователей, можно предписать *Fetchmail* (возможно, с помощью *Procmail*) раскладывать почту по их индивидуальным почтовым ящикам. Тогда каждый сможет получить доступ к своей почте через один и тот же IMAP-сервер (под разными учетными записями, конечно). **НБ**

Краткая справка про...

Использование памяти

Куда девалась память?

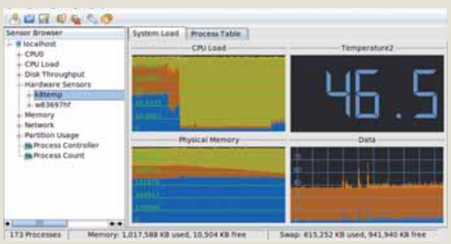
Вы замечали, что через пару часов работы на компьютере счетчик памяти свидетельствует, что свободными остаются лишь несколько килобайт? Вы можете подумать, что нужно добавить памяти, но на самом деле, сколько бы ее ни было, через несколько часов после загрузки (а то и раньше) вся она окажется занятой.

В чем дело? Просто ядро отдает часть памяти, которая не используется процессами, другим нуждающимся, например, кэшу файловой системы, что ощутимо повышает производительность. Взгляните на количество буферов в выводе top и убедитесь, например:

```
Mem: 1028092k total, 1018704k used, 9388k free,
49760k buffers
```

Мы видим, что из 1 ГБ ОЗУ на данном компьютере свободны всего 10 МБ. Половину используемой памяти занимают буферы. Они здорово сокращают время доступа, кэшируя данные в памяти и считывая сверх того, что вам нужно, в предвидении дальнейшего использования. Если память понадобится для чего-то поважнее, например, данных программы, она будет немедленно освобождена.

В памяти остаются не только данные файловой системы. Проследите за ее использованием при запуске и закрытии крупного приложения вроде *Firefox*: вы увидите, что объем свободной памяти уменьшается. Теперь снова запустите *Firefox* и отметьте, насколько быстрее программа загрузилась на этот раз: ядро сохранило данные в памяти для повторного использования. То же относится к общим библиотекам – вот почему смешанный запуск приложений KDE и Gnome не всегда оправдан.



› На вид свободны лишь 10 КБ, но фактически доступной памяти куда больше.

4 Switchdesk или su?

В Инсталлировал Fedora Core 6 и загрузил 200 с лишним обновлений. Хотел выйти из режима администратора, но не нашел команды *switchdesk*. Не переустанавливать же все только из-за этого!

Рон Уайт [Ron White]

О Прежде всего, вообще не следовало запускать рабочий стол от имени администратора! Никакой нужды в этом нет: некоторые дистрибутивы даже намеренно затрудняют загрузку рабочего стола в режиме root. Все программы для системного администрирования можно запускать с рабочего места обычного пользователя. Если понадобятся особые привилегии, у Вас попросят пароль администратора, и сбросят эти привилегии сразу же, как только в них отпадет необходимость. Если Вам понадобилось запустить любую другую привилегированную программу, откройте терминал, станьте администратором при помощи команды **su** – и запускайте ее на здоровье. Это гораздо безопаснее, чем запускать от имени администратора целый рабочий стол (при условии, что вы завершите root-сессию, начатую su, как только в ней отпадет необходимость).

Впрочем, *switchdesk* никуда не делась. Выберите Add/Remove programs (Добавить/Удалить программы) из меню Applications (Приложения) и наберите в строке поиска **switchdesk** – вероятно, вместе с *switchdesk* Вы получите *switchdesk-gui*. Установив программу, можно будет запустить ее через System > Preferences > More Preferences > Desktop Switching Tool (Система > Параметры > Дополнительно > Переключение рабочих »



› В Fedora Core 6 есть *switchdesk*, но не для запуска программ от имени суперпользователя.

столов). Но это не метод запуска программ от имени администратора: *switchdesk* предназначена для того, чтобы пользователь мог переключаться между рабочими столами, отсюда и название. Держите администратора в подобающем месте: заприте его в дальний ящик и извлекайте лишь в случае крайней нужды.

Необходимость в переустановке дистрибутива Linux возникает нечасто. На моем трехлетнем компьютере постоянно работает Linux – система неоднократно обновлялась, но ни разу не переустанавливалась. Переустановка не снимает проблему, она лишь удаляет среду, в которой проблема возникла… пока она не возникнет снова. Если Вы устраните проблему сами, не сметая всю систему, она уйдет навсегда (или, как минимум, ее будет проще устранить в следующий раз). **НБ**

5 Из исходников

В Linux я совсем зеленый, хотя она напоминает мне Amiga прошлых лет. После нескольких попыток, наконец, установил Fedora Core 5 (двойная загрузка с Win XP). Пробовал инсталлировать FreeBasic, следуя вашим инструкциям, но безуспешно! Похоже на то, что не распознается **./configure** и некоторые другие ваши инструкции.

Гордон Оутрим [Gordon Owttrim]

Одно из существенных различий между оболочками Linux и Amiga – Linux не включает текущую директорию в путь поиска по умолчанию, как это делала AmigaDOS. Поведение Linux безопаснее, но тогда при запуске скрипта или программы из текущей директории необходимо указывать полный путь. Текущая директория обозначается как '.', поэтому *./configure* означает «запустить программу или скрипт под названием configure, находящийся в текущей директории». Отсюда ясно, что команда *./configure* срабатывает лишь в том случае, если файл **configure** в данной директории имеется.

Компиляция из исходных текстов обычно включает в себя распаковку архива, переход в каталог, созданный на предыдущем шаге, и запуск *./configure*, сопровождаемый *make* и *make install* – например, вот так:

```
tar xf foo-1.2.3.tar.gz
cd foo-1.2.3
./configure
make
make install
```

Хотя это справедливо для 90% приложений Linux, есть немало исключений. После *cd* загляните в файлы README или INSTALL. Там содержатся специальные инструкции по компиляции и установке данного приложения. Так, если Вы хотите установить *FreeBasic* из исходных текстов, понадобится станцевать *configure – make – make install* несколько раз после загрузки двух архивов. Альтернатива – воспользоваться скомпилированным бинарным архивом, **FreeBASIC-v0.16b-linux.tar.gz**, для которого необходим совершенно иной метод инсталляции, через собственный скрипт. Прочтите **readme.txt** внутри этого архива, чтобы ознакомиться с деталями инсталляции. Настоятельно просим Вас прочесть файл, а не воспроизводить наши инструкции, так как между версиями могут появиться некоторые различия. Файл **readme.txt** следует считать непреложным.

При инсталляции из исходных текстов всегда следует внимательно ознакомиться с инструкциями (в противоположность использованию менеджера пакетов), так как Вы отдаете команды от имени администратора, а это может неблагоприятно отразиться на состоянии системы в случае ошибки. **НБ**

6 ООо, как сложно!

По поводу *OpenOffice.org 2.1* с DVD от LXF90: в Linux я недавно, поэтому не знаю ответа на свой вопрос, да и любой ответ мне, наверное, нужно будет «разжевать». Следуя инструкциям журнала, я попробовал установить OpenSUSE 10.2. Все шло гладко до тех пор, пока я не ввел

```
su -c "rpm -ivh *"
```

что выдало сообщение:

'desktop-integration: not an rpm package (or package manifest): Is a directory'

Тут мои скудные познания и кончились. Проделал то, что казалось мне логичным, и перемещал папку **desktop-integration** повсюду, но ничего не помогало. Пробовал обойти проблему со всех сторон, пока не понял, что просто блуждаю во мраке.

Николас Кейтер [Nicholas Cater]

Если оболочка замечает в командной строке *, то она заменяет эту звездочку на все подходящие файлы – звездочка означает «соответствие любой строке»; в вашем случае, всем RPM-файлам и директории **desktop-integration**. Будьте кон- »

Часто задаваемые вопросы...

Что такое DHCP?

О плетении сетей.

» Я знаю, что DHCP означает Dynamic Host Configuration Protocol (Протокол динамической конфигурации хостов), но что это такое?

Компьютер, подключаемый к традиционной или беспроводной сети, нужно снабдить некоторой информацией, как то: IP-адрес; место, куда следует направлять запросы для неизвестных компьютеров; адреса DNS-серверов. Ручная работа такого рода называется статической адресацией, но для получения таких сведений от сервера компьютер может использовать DHCP. Это существенно облегчает работу в крупных сетях, а также упрощает перемещение от одной сети к другой, с ноутбуками такое часто случается.

» DNS? А что это?

Domain Name Service (Служба доменных имен). Эта служба переводит доменное имя, например, www.linuxformat.co.uk, в сетевой адрес, например 212.113.202.71.

» Как это работает?

Подключившись, клиент – ваш компьютер – рассылает по сети запрос 'discover' (поиск). Запрос получают все компьютеры, но только DHCP-серверы отвечают на него. Каждый сервер посылает в ответ 'offer' (предложение). Клиент выбирает, на какое предложение ответить (если оно не одно), и запрашивает IP-адрес для компьютера, один или больше DNS-серверов, и адрес шлюза.

» Шлюза? Компьютеры сплавлять?

Это не тот шлюз. Стандартный шлюз сети – это адрес, на который направляется трафик для компьютеров чужой сети, таких как Интернет-серверы. Далее шлюз передает трафик по адресу, а ответы пересылает вам.

» А где взять DHCP-сервер?

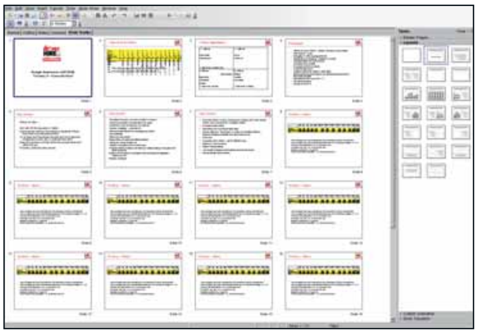
Как правило, для этого не требуется установка отдельного пакета. Большинство маршрутизаторов, включая широкополосные и беспроводные модемы/маршрутизаторы, имеют встроенный DHCP-сервер, который выдает адреса для локальной сети. Интернет-провайдеры тоже пользуются DHCP, когда вы включаете свой компьютер и модем. Если для вашей сети нужен собственный сервис DHCP, то он почти наверняка имеется в вашем дистрибутиве – как сервер, так и клиент. Поищите DHCP в менеджере пакетов. Или возьмите www.thekelleys.org-uk/dnsmasq, которым пользуюсь я.

» Я где-то читал о DHCP leases (арендах) – что это такое?

Когда закрывается сетевое подключение, на DHCP-сервер уходит запрос 'release' (освободить), с помощью которого сервер узнает о том, что выданный адрес больше не нужен и его можно использовать снова. Если же подключение оборвалось неожиданно (погас свет, скажем, или ноутбук вышел из зоны покрытия сети), то сервер не сможет повторно использовать адрес. Время аренды – это тайм-аут для выделенного адреса. DHCP-клиент знает об этом, и до истечения аренды периодически посылает запрос о сохранении подключения. Если клиент вовремя не выслал этот запрос, то адрес может быть передан другому компьютеру.



› DHCP-интерфейс маршрутизатора позволяет назначать точные адреса машинам, необходимые для работы сервера.



› Обновите *OpenOffice.org* с нашего DVD. Только звездочку в оболочке используйте с оглядкой.

» кретнее и используйте

```
su -c "rpm -ivh *.rpm"
```

Теперь это соответствует всему, что оканчивается на **.rpm**, а вам другого и не надо. Если вы хотите инсталлировать RPM-файлы из директории **desktop-integration**, разверните команду следующим образом:

```
su -c "rpm -ivh *.rpm desktop-integration/*.rpm"
```

Заметьте, что простая добавка **desktop-integration/*** не сработает, так как не все файлы в этой директории являются RPM-пакетами, и вы вернетесь к первоначальному сообщению об ошибке. **MC**

7 Модем занемог

В С вашего диска [**LXF87**] установил Mandriva 2007 на Dell 5150. На Dell нет ни параллельного, ни последовательного портов, только USB. У меня Sitecom 56k V.92 USB-модем, модель DC-009, но я не смог подключить к нему Mandriva. USB-мышь и клавиатура работают прекрасно. Командой *modprobe* я загружаю модуль cdc_acm, но он не подключается к **tty**, хотя *KPPP* видит ttyS0 и ttyACM0 и сообщает 'modem busy' [модем занят], когда я запрашиваю модем по ttyS0 или по ttyACM0.

Напрасно я прочесывал Сеть, в том числе www.linux-usb.org. А на Windows XP модем без проблем заработал. Разве эти проблемы не ушли в прошлое?

Мартин Лоуренс [Martin Lawrence]

О Может оказаться, что модем либо не полностью поддерживается, либо требует некоего firmware. Вы не смогли найти сведения о нем в Сети – но это неудивительно. Многие из таких устройств производятся одним изготовителем, а выпускаются под маркой другого. Но чтобы получить право на продажу в США, все они нуждаются в утверждении FCC (Federal Communications Commission, Федеральной комиссии по связи), поэтому узнать о том, что они из себя представляют, можно по их ID-коду FCC. Введите код в поле внизу страницы www.hardwaresecrets.com/page/fcc, и вы узнаете, кем изготовлен Ваш модем. Подкрепленный добытой таким образом информацией, поиск на сайтах вроде www.linux-usb.org окажется гораздо более плодотворным.

Еще одна сложность вашей ситуации – USB-модемы недостаточно стандартизованы, лишь немногие из них соответствуют спецификации CDC-ACM. Ваши клавиатура и мышь хорошо работают именно потому, что отвечают требованиям стандарта (USB HID). Один из вариантов обхода проблемы состоит в использовании последовательного модема через специальный USB-адаптер. У меня есть пара таких штучек, с разными чипсетами (обе недорого куплены на eBay), и они хорошо работают со всеми последовательными устройствами, которые я пробовал (за исключением ИБП). Таким способом можно использовать не только модем, но и любое необходимое Вам последовательное устройство. Мы-таки избавились от параллельного порта и трехдюймового дисковода, но старый последовательный порт упорно не желает сдаваться. **НВ**

8 Вопрос Qemu

В Меня интересует эмулятор *Qemu*, но он не работает так, как описано в вашем журнале. Сообщает, что ему нужен *GCC 3*, а на OpenSUSE 10.2 есть только *GCC 4*. Могу я узнать, опробует ли хоть кто-то еще из вашего коллектива различные ваши программы и идеи, прежде чем публиковать их в журнале?

Джозеф Лэм [Joseph Lamb]

О *Qemu* – одна из очень немногих программ, пока не работающих с *GCC 4*, и досадно, что новые дистрибутивы вроде OpenSUSE больше не включают пакетов *GCC 3*. Но Вы можете установить *GCC 3* как из исходных текстов, так и из пакетов от Fedora Core 5, которые, по нашим сведениям, работают с OpenSUSE 10.2.

Однако для единственной программы это слишком трудоемко, проще найти готовые пакеты. Один из пользователей OpenSUSE 10.2 скомпилировал *Qemu* и выложил его по адресу www.hasanen.com/files/linux/qemu.tar.gz. Есть пакеты и на сайте SUSE. Укажите браузеру на http://download.opensuse.org/distribution/SL-OSS-factory/inst-source/suse/i586 и щелкните на файле *Qemu* (текущее название **qemu-0.9.0-3.i586.rpm**, но к моменту выхода журнала оно может измениться). В ответ на вопрос браузера о том, как поступить с этим файлом, выберите «инсталлировать» и «дождаться, пока файл будет загружен и инсталлирован» (введите пароль администратора, когда потребуется).

Можно установить *Qemu* и из командной строки:

```
su -c "rpm -ihv http://download.opensuse.org/
distribution/SL-OSS-factory/inst-source/suse/i586/
qemu-0.9.0-3.i586.rpm"
```

Мы всячески заботимся о достоверности нашей информации, не исключая тестирования инструкций разными людьми. Но нет никакой гарантии, что любое действие сложнее echo Hello World безупречно сработает на любой конфигурации, и это печальное следствие огромного количества дистрибутивов и бесконечных мутаций различных пакетов на них. Добрая весть – новые версии *Qemu* должны быть совместимы с *GCC 4*. **MC**

9 cron: чистит до блеска

В Устав вычищать следы спам-атак на мой сервер, я задумал построить *cron*-процесс для автоматизации работы, но не могу найти подходящей команды для этого. Буду благодарен за любой комментарий к моему коду:

```
mysql -h hostname -u username -ppassword -e'delete
from table_name where pn_cid > x'
```

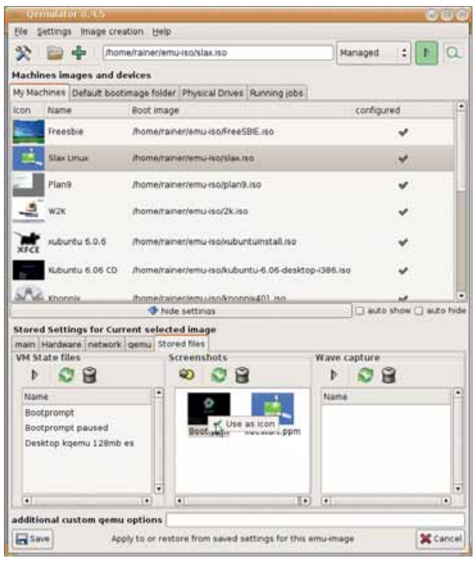
! Вопрос-победитель (русская версия)

Алексей Николаевич получает подарочный сертификат на 1000 рублей от интернет-магазина LinuxCenter.Ru! Просим победителя выйти на связь с редакцией: info@linuxformat.ru

В В настоящий момент у меня есть только одна проблема, которая мешает переводу всего и вся под Linux. Это автоpазмонтирование съемных носителей. Люди, привыкшие к работе с Windows, не видят ничего зазорного в выдергивании дискет и «флешек» в любой момент времени. Это приводит к печальным потерям данных. Есть ли способ приблизиться к решениям Microsoft и в этом вопросе?

Виталий Токаренко

О Вам может помочь опция sync команды *mount*, которая предписывает ядру сохранять данные мгновенно, а не через некоторый промежуток времени, как это обычно делается. С ней вы можете быть уверены, что к тому моменту, когда погаснет индикатор записи на устройстве, все данные будут зафиксированы в постоянной памяти. Таким образом можно сократить потери информации при внезапном выдергивании носителя, а заодно и срок его жизни, если речь идет о «флешках» (как известно, они выдерживают вполне определенное число циклов чтения/записи, а механизм кэширования, реализуемый ядром, позволяет избежать ненужных обращений). Кстати, выдергивать USB-брелок, на который только что производилась запись, не рекомендуется даже в Windows: там для этого есть специальный пункт в контекстном меню – «Безопасное извлечение устройства». **ВС**



› Для компиляции *Qemu* требуется *GCC 3*, но у большинства дистрибутивов имеются готовые пакеты.

Вопрос-победитель (английская версия)

Прямиком в тюрягу

Я хочу так настроить мою систему SUSE 10.2, чтобы пользователи подключались через *OpenSSH*, а я, даже подключившись в качестве обычного пользователя, мог бы видеть любую папку системы. Как можно ограничить доступ таким образом, чтобы пользователи видели лишь папки внутри домашней директории, которую я создаю для них при настройке учетной записи? Можно ли оставить им лишь функции SFTP или SCP?

Кевин Винтер [Kevin Winter]

SSH-вход виртуально не отличается от обычного локального входа, только действует через зашифрованный туннель, и подключающиеся через SSH пользователи обладают теми же правами, которые они имеют, сидя за вашим компьютером. Обычно это означает, что они могут читать системные директории (иначе они не смогут запустить ни одной программы) но не изменять их. Можно настроить систему на chroot пользователя к его домашней директории при входе, но это далеко не простая задача. Если вы все же возьметесь за ее решение, то, по-моему, вам пригодится набор программ, способных упростить это занятие: *Jailkit* [jail – тюрьма] с http://olivier.sessink.nl/jailkit.

С помощью *Jailkit* можно также ограничить подключение пользователей только SFTP или SCP, но если это все, что Вам надо, было бы проще применить программу *scponly* (http://sublimation.org/scponly). Она замещает оболочку и запрещает обычный вход, но принимает SFTP- и SCP-подключения. Простейший путь ее использования – заменить оболочку пользователя на *scponly* в **/etc/passwd**. Это предотвратит вход через оболочку, но позволит передвигаться по файловой системе в соответствии с правами доступа к папкам и файлам. Параметр **chroot** для *scponly* тоже имеется, но это на порядок усложняет задачу – если вам без этого не обойтись, лучше уж придерживаться *Jailkit*.

Трудность настройки **chroot**-входа состоит в том, что нужно обеспечить пользователю нормальную работу внутри домашней директории и одновременно запретить читать то, что ему не полагается. Программы вроде *Jailkit* в некоторых ситуациях хорошо справляются с такими задачами – например, при настройке **chroot** для сервера – но с интерактивными подключениями они работают гораздо хуже. В зависимости от ваших требований к безопасности, в большинстве случаев лучше укрепить систему на локальном уровне, скрыв критические системные файлы от не-администраторов, а затем распространить эту настройку на все SSH-, SFTP- и SCP-подключения. **НБ**

На моем сервере не одна база данных, а строка моего кода не содержит упоминания об использовании конкретной базы – так, может быть, это холостой выстрел?

SwissMac, из форумов LXF

Указывать название базы данных все-таки необходимо (пусть она у вас и одна), иначе *MySQL* не будет знать, как применить команду. Название базы можно передать либо в качестве последнего параметра командной строки, либо среди других команд, передаваемых *MySQL*. Вот эквиваленты:

```
mysql -u user -ppword -e "delete from table_name where pn_cid > x" database_name
mysql -u user -ppword -e "USE database_name; delete from table_name where pn_cid > x"
mysql -u user -ppword - "delete from database_name.table_name where pn_cid > x"
```

Первый вариант проще, зато остальные более гибкие. Можно передавать команды клиенту *MySQL* конвейером вместо использования параметра **-e**. Это полезно, если нужно запустить несколько команд, так как можно поместить их в файл и делать так:

```
mysql -u user -ppword <cmdfile
```

Следует учесть, что программы, запускаемые в *cron*, попадают в среду, отличную от установленной в вашей оболочке. Имя пользователя отсутствует, поэтому переменные окружения, перечисленные в вашем профиле, не настраиваются. Это можно обойти, поместив следующие команды в небольшой скрипт:

```
#!/bin/sh
source /etc/profile #или ~/bash_profile или ~/.bashrc
mysql -uuser -ppword ...
```

Укажите файл, содержащий настройки вашей среды, во второй строке, и настройте *cron* так, чтобы он вызывал скрипт вместо непосредственного запуска *MySQL*. Использование скрипта немного облегчает тестирование, так как есть уверенность в том, что используемые команды действуют одинаково как в терминале, так и из-под *cron*.

Помещение пароля в командную строку считается небезопасным, так как при работе программы любой пользователь машины может узнать пароль, просто взглянув на вывод *ps*. Безопаснее будет поместить пароль в **~/.my.cnf**, вот так:

```
[client]
password=your_pass
```

и затем выполнить

```
chmod 600 ~/.my.cnf
```

Теперь этот файл сможет прочесть лишь Ваш пользователь. **НБ**

10 От меня к тебе

Как изменить адрес 'from' (от) при использовании Linux-команды *mail*? Она настаивает на пометке сообщения как user@user-laptop (user-laptop – это мое имя хоста).

hubris, из форумов LXF

Со стандартной командой *mail* это невозможно без игры с переменными окружения **USER** и **HOSTNAME**, которая может повредить другим программам, работающим в той же оболочке. Зато есть много других программ, способных исполнить Ваше желание. Если у Вас установлена *Mutt*, то она может взять адрес «от» из переменной окружения **EMAIL**. Но если нет, и Вам нужно лишь настроить отправку сообщений, Mutt чересчур громоздка. Альтернатива попроще – *SMTPClient* (www.engelschall.com/sw/smtpclient), в работе она напоминает *mail*, но принимает аргумент --from для установки адреса «от». *SMTPClient* отправляет сообщения на соответствующий SMTP-сервер, по умолчанию – на localhost. Если нужен другой сервер, то его имя можно указать в параметре командной строки **--smtp-host**, а можно настроить переменную окружения **SMTPSERVER**.

```
echo "Hello world - what else?" | smtpclient
  --smtp-host=my.mail.server
  --from=hubris@wherever
  --subject «Hello World»
  someone@someplace
```

ПХ

11 Спам достал

У меня небольшая офисная сеть со своим почтовым сервером, на котором работает *Sendmail* в паре со *SpamAssassin*. *Fetchmail* забирает почту со всех POP3-серверов, затем она раскладывается по папкам пользователей, откуда они загружают ее к себе через *qpopper*. Вопрос касается *SpamAssassin* (у меня стоит последняя версия, т.к. я постоянно обновляю ее через CPAN): я хочу добиться того, чтобы письма от определенных отправителей (whitelist) никогда не воспринимались как спам, независимо от их содержания. Но *SpamAssassin* работает по принципу начисления «очков», а опция *whitelist* дает их слишком мало для того, чтобы повлиять на окончательное «решение». Я хотел было изменить принятое по умолчанию число очков для *whitelist*, но это оказалось невозможным. Получается, если члены списка *whitelist* отправят письмо и оно будет идентифицировано как спам, то whitelist «не помогает». Нет ли каких-то обходных путей? Я хочу добиться, чтобы *SpamAssassin* «пропускал» указанных мною отправителей или уж пускай вообще не проверяет письма от них. Какие возможны решения?

Константин

Согласно описанию, опция **whitelist_from** дает коэффициент -100, что само по себе более чем достаточно – надо очень сильно постараться, чтобы легитимный клиент набрал более чем 100 баллов и был принят за спам. Для *SpamAssassin*, настроенного на индивидуальных пользователей (**~/.spamassassin/user_prefs**) можно указать опцию

```
whitelist_from d.cary@sparkingwire.com
```

Для общей настройки следует проделать то же самое в **/etc/mail/spamassassin/local.cf** (в зависимости от системы путь может меняться). Альтернативный метод – написать собственные правила вида:

```
header NOSPAMFROM From =~ /notaspamer\@example\.com/i
score NOSPAMFROM -100.0
```

Подробности можно найти по адресу http://wiki.apache.org/spamassassin/ManualWhitelist.
Андрей Тусеев

12 Бесплатный сыр...

В С тех пор, как у нас появился безлимитный GPRS-интернет, я постоянно ставлю телефон (в связке Bluetoth + KPPP) на закачку на ночь. Но иногда демон PPP «вылетает» с ошибкой, а так как я в это время обычно сплю, то обнаруживаю это только утром. Можно ли сделать автоматическое восстановление соединения при его разрыве или крахе демона ?
Казанцев А.С.

О Конечно, можно. Однако, способ решения этой задачи зависит от первопричины – т.е. от того, из-за чего именно «вылетает» pppd. Если соединение (случайно или намерено) обрывает оператор, можно воспользоваться средствами самого pppd – т.е. опциями persist (не выходить при обрыве соединения, пытаться перезвонить), holdoff (время ожидания перед следующей попыткой дозвона) и maxfail (число неудачных попыток, после которых демон все же завершит работу). Все они описаны в man и вы можете передать их (вместе с соответствующими параметрами) посредством KPPP – для этого в окне свойств учетной записи предусмотрена специальная кнопка «Customize pppd Arguments».Если же обрыв происходит локально (например, прерывается Bluetooth-канал или «подвисает» телефон, что тоже случается), придется писать специальный скрипт, который проверяет доступность аппарата и, в случае необходимости, производит повторную инициализацию. **Валентин Синицын**

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.



Большой вопрос

Раздел Ответов закончен, но наши эксперты не уходят домой, пока хоть один не проникнет в суть общей проблемы

13 Избавляемся от проводов

В Я загрузил и установил OpenSUSE 10.2, все работает прекрасно, кроме моей беспроводной сети. Я новичок, и хотел бы найти руководство по настройке для начинающих.

frostyuk, из форумов LXF

О Ответ зависит от чипсета, на котором собрана беспроводная карта. Сначала запустите *Yast* и перейдите в раздел сетевых карт. Если Ваша карта отображается, переходите к следующим шагам; в противном случае карту нужно идентифицировать, набрав *lspci* или *lsusb* (в зависимости от типа карты) в терминале. Можете ввести полный путь к *lspci*, **/sbin/lspci**.

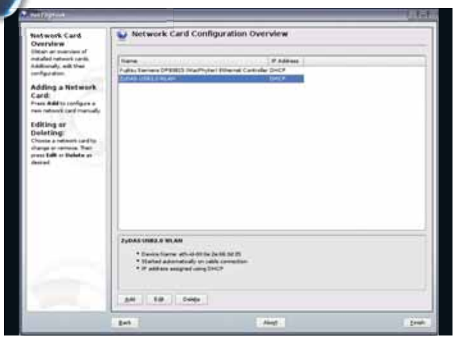
Поиску драйвера для карты поможет Google. Большой список USB-устройств и драйверов для них имеется на **www.qbik.ch/usb/devices**. Узнав, какой драйвер Вам нужен, убедитесь, что он установлен: некоторые драйверы включены в ядро Linux, некоторые нужно устанавливать отдельно. Наберите **/sbin/modprobe --l** в терминале для просмотра списка «ядерных» драйверов. Если нужного Вам там нет, воспользуйтесь полем поиска в разделе управления программами *Yast*. Не обнаружив нужного драйвера через *Yast*, перейдите на домашнюю страницу драйвера и следуйте инструкциям по его сборке из исходных текстов. Если драйвер в системе присутствует, а карта не распознается, то Вам, вероятно, нехватает файла прошивки (firmware). Обычно они извлекаются из драйверов для Windows. В этом случае тоже нужно следовать инструкциям с сайта драйвера.

Например, *lsusb* определяет мое USB-устройство Edimax как устройство ZyDAS. Вышеуказанный сайт подтверждает, что оно использует драйверы zd1211 (включенные во все новейшие ядра), но для устройства необходимы также прошивки, доступные на **http://sourceforge.net/project/showfiles.php?group_id=129083**. SUSE считает, что прошивка находится в **/lib/firmware/zd1211** (другие дистрибутивы могут использовать просто **/lib/firmware**), поэтому создайте эту директорию, распакуйте архив с прошивкой и скопируйте все файлы в **/lib/firmware/zd1211**.

Теперь вернитесь в раздел сетевых карт *Yast* – а если ваша карта уже была опознана, пора браться за дело. Беспроводное устройство должно появиться в списке; выберите его и нажмите кнопку Edit (Изменить). На вкладке Address (Адрес) выберите Automatic Address Setup (Автоматическая установка адреса), а на вкладке General (Общие) установите Device Activation (Активацию устройства) на On Hotplug (При подсоединении), если это сменное USB-устройство или имеется выключатель (тогда беспроводная сеть создастся при подсоединении или включении устройства); иначе, пользуйтесь настройкой Manual (Ручной) и управляйте устройством через значок менеджера сети в панели задач. Нажмите Next (Далее) и Finish (Готово) для выхода из настройки.

Перейдите к менеджеру сети в панели задач, он должен предъявить список доступных сетей. Выберите свою. Если сеть защищена WEP- или WPA-шифрованием, будет запрошен пароль. Постарайтесь снять шифрование на время настройки подключения: пока устройство не заработало, смысла в нем нет, есть только усложнение! Как только сетевое подключение состоится, отключитесь, включите шифрование и подключитесь снова. Система спросит у Вас пароль WEP или WPA, и подключение станет безопасным. **НБ** LXF

Шаг за шагом: Настройка беспроводного подключения



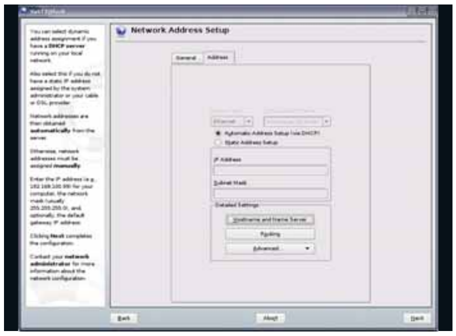
1 Поиск устройства

Если устройство обнаружится в разделе сетевых карт *Yast*, то вам повезло; если нет, потрудиться придется побольше.



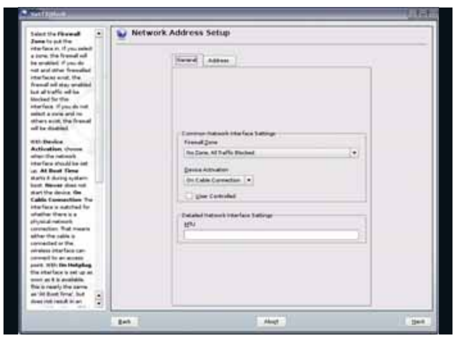
2 Поиск драйвера

Если устройство не отображается, введите его название в поле поиска на **www.qbik.ch/usb/devices** для поиска требуемого драйвера.



3 Назначение IP-адреса

Беспроводные устройства, как правило, используют DHCP, так как легко перемещаются между сетями. Назначайте адреса вручную только при крайней необходимости.



4 Активация устройства

Выбор On Cable Connection (При подсоединении кабеля) для беспроводной сети странноват, но зато находит точки доступа. Для сброса автоматического подключения выберите ручное управление.



5 Шифрование

Настройка подключения упростится, если вы временно отключите шифрование на своей точке доступа. Не забудьте потом сразу же вернуть его.



6 Проверка подключения

Удачное завершение! Окно информации о подключении менеджера сети сообщает, что все работает, как положено.



LXF DVD93

Linux для предприятий всего в нескольких кликах мыши



Майк Сондерс любовно компонует диск Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

Всерьез и надолго...

Термин «стабильный дистрибутив» вызывает немало разночтений. Многие новички в Linux вполне естественно считают, что он подразумевает надежный и в основном не имеющий ошибок дистрибутив, но на самом деле его значение более глубокое: дистрибутив, который не изменяется. Доморощенные хакеры часто высмеивают стабильную ветвь Debian из-за того, что она содержит старое ПО, и единственное, что обновляется – это безопасность (вместо функциональности); однако для бизнеса подобный подход весьма разумен.

Если вы развернули Linux на 1 000 серверов и на 30 000 рабочих столов, то вам не нужны версии программного обеспечения, обновляемые каждые пять минут по причине того, что разработчикам заблагорассудилось обновить репозитории дистрибутива. Ошибки сами по себе не являются серьезной проблемой – всегда можно найти обходной путь; но оставить вечером Firefox 1.x и придти наутро к обновленной версии 2.x на всех рабочих столах невозможно без привлечения многочисленного персонала. Придется заново проверять web-приложения, приучать пользователей к изменениям интерфейса и, естественно, исправлять новые ошибки и решать новые проблемы.

Вот поэтому-то и следует уважать «стабильные» дистрибутивы, такие, как Debian, RHEL, CentOS и StartCom Linux. Что ни день, лазая по Интернету, я постоянно встречаю людей, ругающих Debian или CentOS за то, что у них нет самой последней версии всего. Они не понимают, что их козырь – долгосрочная стабильность. Так давайте воздадим этим дистрибутивам заслуженные ими честь и славу, и отметим тот факт, что Linux в состоянии учесть все потребности! Познакомьтесь с StartCom EL 5 на нашем DVD – может, у него и нет свежайших версий каждого пакета, но он более чем тщательно протестирован и, чтобы вы знали, не рухнет внезапно в связи с новым витком обновлений.

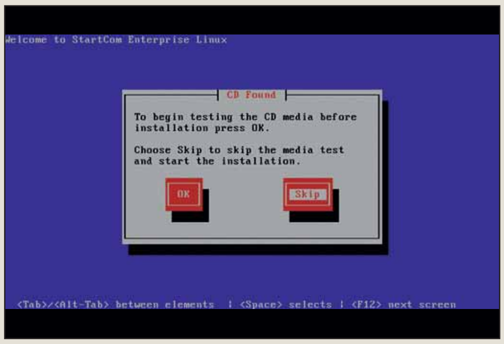
mike.saunders@futurenet.co.uk

Шаг за шагом: Устанавливаем StartCom EL 5



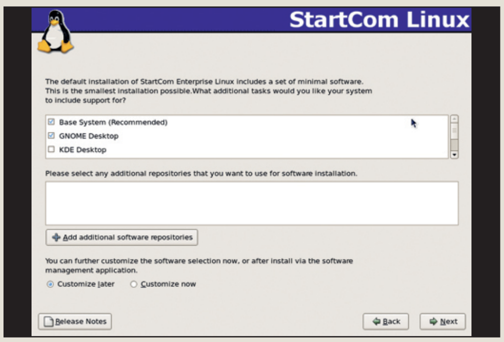
1 Загрузка

Загрузите компьютер с LXFDVD (возможно, для этого вам понадобится изменить настройки BIOS – установить загрузку с DVD, а не с жесткого диска). Введите «text» на машинах, где установлено 128 МБ ОЗУ или менее.



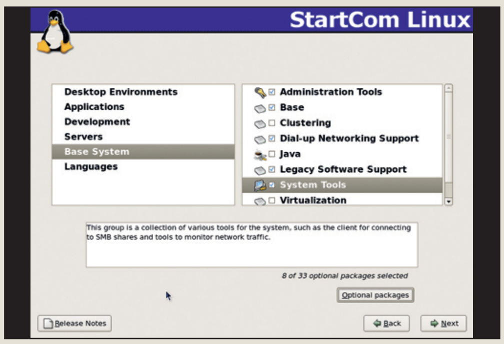
2 Мимо теста

Программа установки может начать проверку DVD на наличие ошибок. Пропустите этот шаг (он займет некоторое время), в случае возникновения проблем это действие можно запустить позже.



5 Выбор ПО

Установите пароль для администратора (запомните его!), затем укажите Base System, Gnome, Office, Productivity и Administration Tools на стадии выбора ПО.



6 Сверх обычного

Нажмите кнопку Customize Now внизу, затем – Next, и выберите любые дополнительные пакеты, которые вам нужны.

Дистрибутив Linux

StartCom EL 5

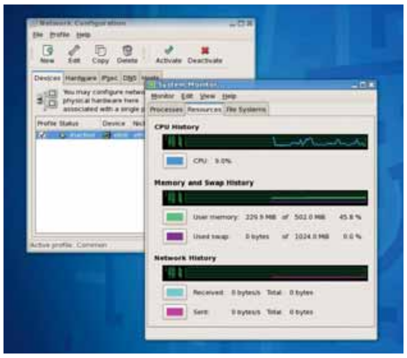
Red Hat с уважением относится к философии свободного ПО, не считая его лишь способом быстро заработать. Лучший пример тому – RHEL, дистрибутив Linux для предприятий. По условиям лицензии GPL, Red Hat обязана предоставлять доступ к своему исходному коду, поэтому в том, что фирма выложила заплатки в Сеть для скачивания, никакого альтруизма нет. Однако кое-что Red Hat все-таки добавила, и это – SRPM-пакеты для корпоративного продукта. То есть и вы, и я, и любой другой может скачать их и воссоздать те самые двоичные пакеты, которые Red Hat использует в своем коммерческом продукте – с теми же самыми зависимостями, флажками компиляции и наборами заплаток.



› **Желаете 3D-куб *Compiz*? Активируйте Desktop Effects в System Preferences.**

Естественно, воссозданные проекты не могут использовать логотипы или товарные знаки Red Hat, и они должны четко указывать, что их программы не являются поддерживаемым продуктом Red Hat, но во всем остальном они отлично работают. Крупные фирмы, которым нужны договоры на поддержку, приобретут оригинальный продукт у Red Hat, а домашние пользователи или любители могут наслаждаться прочным как скала дистрибутивом Linux, не тратя сотни долларов на поддержку, которой они никогда не воспользуются.

В прошлом выпуске мы рассматривали RHEL 5, массивный корпоративный релиз, содержащий *Xen*, *Compiz* и прочие современные технологии. Тестирование показало его быстроту и стабильность. Уже через неделю после выхода релиза сообщество занялось пересборкой RHEL 5 из исходных текстов, как было описано выше, и StartCom первым выложил готовый продукт. Поэтому мы с гордостью представляем на DVD этого месяца StartCom Enterprise Linux 5, отменно протестированный и отполированный дистрибутив для дома или офиса. Подобно RHEL, StartCom идеально подходит и для рабочих станций, и для серверов, и если вам нужен надежный дистрибутив, который не рухнет, вы не ошибетесь с выбором.



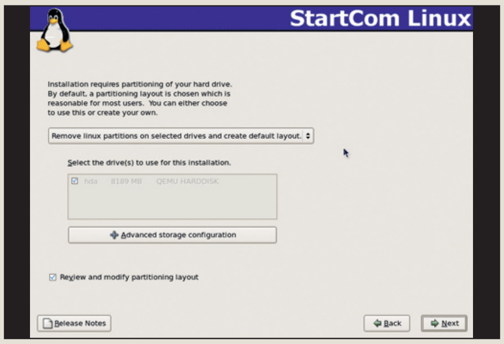
› **StartCom использует знакомые инструменты настройки system-config-***

Эта версия StartCom EL 5 разработана для 32-битных x86-ых ПК (но она будет работать на AMD64) с 256 МБ ОЗУ и процессором как минимум 1 ГГц для достойной производительности рабочего стола. Если на вашем компьютере меньше памяти, вам, возможно, удастся установка в текстовом режиме. Ниже вы найдете руководство по инсталляции, а через страницу – подробности работы с дистрибутивом. Если у вас есть представление об использовании Linux, то никаких проблем с ним у вас не возникнет.

Установив StartCom EL 5 при помощи нашего руководства на предыдущих страницах, вы возможно захотите исследовать дистрибутив и узнать, как же он работает. Если вы раньше использовали Fedora Core, мно- »

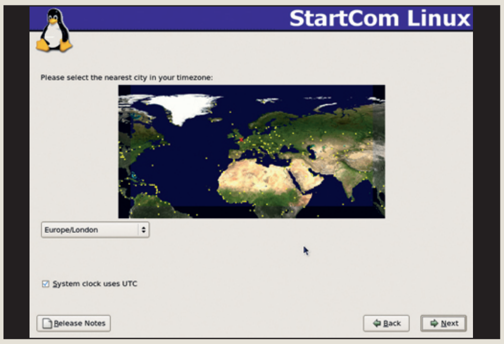
Учтите

Хотя StartCom собран из исходных кодов RHEL 5, это совершенно отдельный продукт, и он не поддерживается Red Hat. Загляните на www.startcom.org, там вы найдете справочные форумы и информацию.



3 Место на диске

Можете отдать StartCom полностью весь жесткий диск, а можете выбрать Create Custom Layout, чтобы создать swap-раздел объемом 512 МБ и выделить не менее 5 ГБ под корневой раздел (/).



4 Место на карте

Установите *Grub*, настройте свою сетевую карту и укажите свое географическое положение (и тем самым часовой пояс).



7 Ждите установки

Программа установки StartCom отформатирует ваши разделы Linux и начнет копировать пакеты. Обычно это занимает от 20 минут до часа.



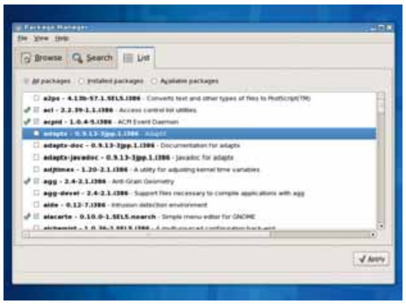
8 Финиш

После окончания установки программа перезагрузится и нанесет последние штрихи (например, создаст учетную запись обычного пользователя). И все готово к работе!

» гие аспекты будут вам знакомы – например, структура файловой системы и настройки оболочки: в конце концов, именно Fedora послужила основой для релизов RHEL. Рабочим столом по умолчанию в StartCom является Gnome 2.16, в компании с *OpenOffice.org 2.0.4* и *Firefox 1.5.0*; все это работает на ядре 2.6.18, *Glibc 2.5* и *X.org 7.1*.

Как отмечалось в предисловии к статье о содержании DVD, номера версий не всегда самые последние из имеющихся, зато все программы очень тщательно протестированы. Вы также увидите, что исправления ошибок и брешей в безопасности включаются в выпущенную версию пакета – например, если в Apache 2.2.5 обнаружена уязвимость, исправленная командой Apache в версии 2.2.6, то команды разработки RHEL и StartCom обычно включают исправление ошибок в свою версию *Apache 2.2.5*. Это означает, что в вашем дистрибутиве не появится никаких неожиданных изменений или новых функций, когда вы обновите его; у вас просто появятся исправления ошибок, а все остальное останется стабильным.

Как и следовало ожидать, StartCom пользуется пакетом инструментов настройки *system-config-** от Fedora и RHEL. Они позволяют сконфигурировать сеть, принтеры и прочие аспекты системы без мучений с командными строками. Вы увидите, что StartCom 5 включает SELinux, систему безопасности, позволяющую ограничивать программный доступ к ресурсам (например, к файлам, оборудованию и т.д.), и при нарушении безопасности вы сможете изолировать преступника (*см. подробности на стр. 46*).



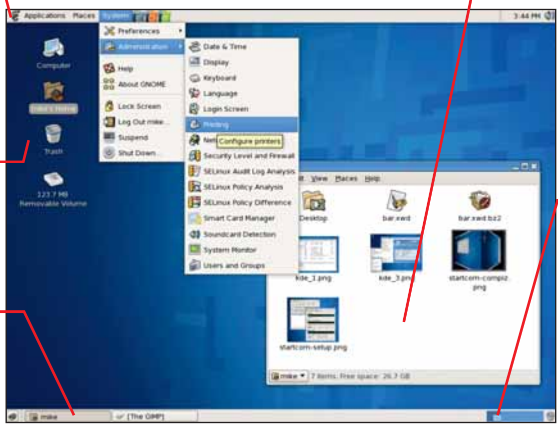
› **Просмотрите интернет-репозитории пакетов StartCom через *Pirut*.**

Исследуем рабочий стол StartCom

Меню
Нажмите на Applications, чтобы запустить программы, Places – чтобы просмотреть файловую систему, и System – для настройки.

Значки
Когда вы добавляете новое устройство, например, USB-накопитель, на рабочем столе появляется такой значок.

Панель задач
Как и следует ожидать, она просто позволяет вам переключаться с одной работающей программы на другую.



Nautilus
Пространственная навигация, любите вы ее или нет, поставляется по умолчанию с StartCom'овской версией Gnome.

Рабочее пространство
Четыре виртуальных рабочих стола позволяют выбрать рабочее пространство для инструментов Интернет, офисных приложений и т.п.

Если вы захотите установить программу, то по умолчанию StartCom будет искать пакеты в Интернете. Например, чтобы установить *Zsh* при наличии доступа к Сети, введите

```
yum install zsh
```

от имени администратора в строке приглашения оболочки. Если вы не подключены к Интернету и хотите установить пакеты с DVD, понадобится настроить локальный репозиторий. Вставьте LXF DVD, и если не начнется автоматическое монтирование, введите (опять же, в режиме администратора)

```
mkdir /mnt/cdrom
mount /dev/cdrom /mnt/cdrom
```

Затем нужно установить инструмент *createrepo*, чтобы создать новый репозиторий, и скопировать RPM пакеты на нашу локальную машину:

```
rpm -Uvh /mnt/cdrom/StartCom/RPMS/createrepo*
mkdir /root/RPMs
cp /mnt/cdrom/StartCom/RPMS/* /root/RPMs
umount /mnt/cdrom
cd /root/RPMs
createrepo .
```

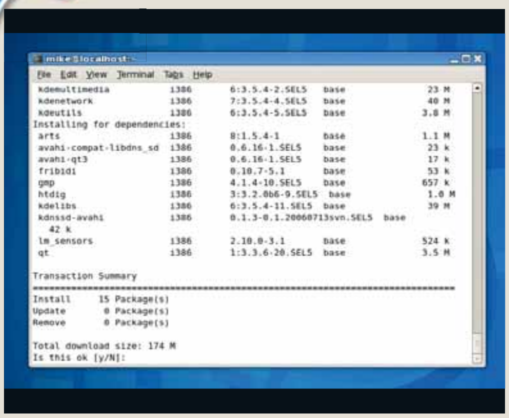
Мы скопировали локальный RPM-репозиторий с DVD, в **/root/RPMs**. Чтобы им воспользоваться, нужно настроить *Yum*, поэтому откройте **/etc/yum.repos.d/startcom.repo** (сперва неплохо бы сделать резервную копию!), удалите его содержимое и введите

```
[base]
name=Base
baseurl=file:///root/RPMs
enabled=1
gpgcheck=0
```

Теперь, когда вы введете команду **yum install zsh**, она возьмет *Zsh* из локального репозитория и не будет лазить в Интернет. Если у вас появится Интернет-соединение, просто скопируйте старый файл **startcom.repo** (резервную копию которого вы, как мы надеемся, сделали) назад, на место.

Ждите DVD следующего месяца, на котором будут обновления StartCom!

Шаг за шагом: Переходим с Gnome на KDE



1 Установка
Если вы не выбрали KDE во время установки, зайдите в Applications >Accessories >Terminal, введите **su** для входа в сеанс администратора, а затем введите **yum install kdebase kdegraphics kdenetwork kdemultimedia**



2 Вход
Откройте System > Log Out имяпользователя, затем в окне приглашения нажмите Session и выберите KDE. Теперь войдите в систему как обычный пользователь.

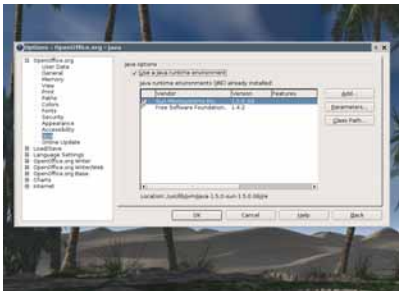
Офисный пакет

OpenOffice.org 2.2

В продолжение борьбы с *MS Office* появился *OpenOffice.org 2.2*, дабы доказать, что предложение Microsoft вовсе не представляет собой всеобъемлющего и непревзойденного офисного ПО. Конечно, OOo склонен к медлительности и утяжеленности, но в нем потрясающе много функций – и некоторых из них (например, встроенного экспорта в PDF) явно не хватает в *MS Office*. Команда OOo ускорила темп появления релизов, выпустив версию 2.2 всего через четыре месяца после версии 2.1 (которая появилась аж через год после 2.0).

Уплотнение релиз-цикла – ответ на жалобы пользователей: люди не хотят ждать год или два, чтобы на них свалилась огромная масса функций. Сказалась и необходимость оставаться в колонках новостей – в особенности с учетом того объема, который уделяется в них *Office 2007* на данный момент.

Итак, что нового в 2.2? Первым делом в глаза бросается обработка шрифтов с поддержкой кернинга (настраиваемого межсимвольного расстояния), активированного по умолчанию. Обновлена функция экспорта PDF – теперь она включает закладки и поля формы, устанавливаемые пользователем, а



› Подсказка: отключите Java (через Tools > Options), и запуск ускорится.

совместимость с Microsoft продолжает совершенствоваться, благодаря улучшению поддержки сводных таблиц в *Calc*. Как обычно, этот релиз исправляет немало ошибок.

Функции и хороши, и милы, но особенно вам понравится рост производительности. Холодный старт *OOo Writer 2.2* на тестовом 1.6-ГГц компьютере на две секунды быстрее, чем в 2.1 – по-прежнему довольно медленно при десяти секундах в целом, но все же и это очень важное достижение. Майкл Микс [Michael Meeks] из Novell старательно трудится над улучшением производительности, а другие разработчики начали устранять утечки памяти. Посмотреть список всего, что стало быстрее работать в 2.2, можно на http://tinyurl.com/23l9qn.

Для установки этого релиза, сначала удалите все предыдущие версии с помощью менеджера пакетов, затем скопируйте **openoffice.org-2.2-linux.tar.gz** из раздела **Office/OpenOffice.org** нашего DVD в домашнюю директорию. Распакуйте этот архив и установите пакеты в папку **RPMS** полученной директории. Например, в системе на базе RPM типа Fedora или SUSE надо открыть терминал, войти (*cd*) в **RPMS**, ввести **su** для запуска сеанса администратора, а затем набрать:

```
rpm -ivh *.rpm
```

Если на вашей машине установлены Debian или Ubuntu, вам придется конвертировать эти пакеты в Debs таким образом (от имени администратора):

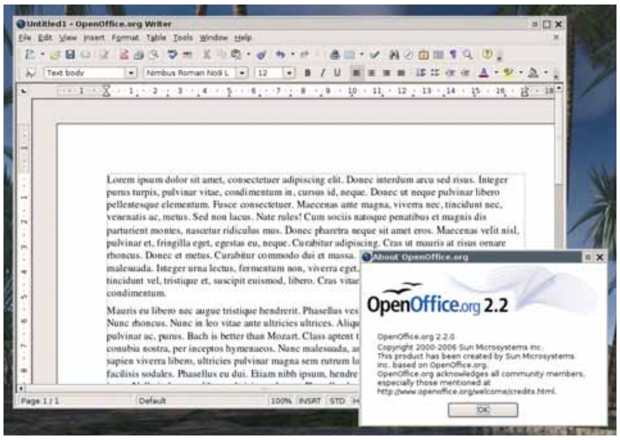
```
alien *.rpm
dpkg -i *.deb
```

OOo 2.2 будет установлен в **/opt/openoffice.org2.2/**, и вы можете запустить его, введя команду

```
/opt/openoffice.org2.2/program/soffice
```

или создав соответствующий ярлык на рабочем столе.

»



› Добрые вести: *OpenOffice.org* запускается чуть быстрее, и выглядит лучше благодаря новым чарам кернинга.



3 Работа

Вы попадете на рабочий стол KDE. Если хотите вернуться снова в Gnome, просто снова измените Session на экране приглашения Шага 2.

Документация

Журнал в формате PDF

Как обычно, в журнале не только те страницы, которые вы читаете в данный момент. На нашем DVD, в разделе **Magazine/PDFs** вас ждут целых три статьи из предыдущих номеров *Linux Format*, доступные в любой программе для чтения PDF (например, *Evince* в Gnome). Мы начинаем со статьи Строим Linux-киоск из LXF79, где более чем на шести страницах рассказывается, как создать компактную целевую Linux-систему.

Для многих музеев, магазинов и даже Интернет-кафе компьютер с полнофункциональной Linux/Windows – явный перебор; вероятно, для работы будет достаточно одной-двух программ (в наши дни это чаще всего web-браузер). Поэтому мы показываем, как избежать излишеств, подобрать нужное оборудование для выполнения этой задачи и установить Linux таким образом, чтобы пользователи ничего не сломали.

Из LXF81 мы взяли исследование Йона Филипса [Jon Phillips] о Creative Commons. Всем нам известны преимущества Open Source, но когда дело доходит до предоставления во всеобщее пользование музыки и произведений искусства и литературы, вопросы прав и их нарушения становятся более туманными. Йон объясняет, как Creative Commons облегчает людям продвижение своих работ и возможность сделать их общедоступными без риска быть ограбленным или дискредитированным.

И, наконец, мы заделались любезными технарями, приведя руководство из LXF82 по размещению вашей домашней директории в *Subversion*. Звучит странновато, правда? Но, используя систему контроля версий для своих персональных файлов, вы можете мгновенно вернуться к предыдущим редакциям документов и восстановить случайно удаленные файлы. Вы поняли, что те три параграфа, которые вы выкинули из документа на прошлой неделе, все-таки нужны? *Subversion* вас выручит.



› Есть что показать всем в сети? Берите лицензию CC.

Другие программы

Новые релизы

На DVD этого месяца еще много с чем стоит познакомиться, включая нашу подборку самых лучших и самых интересных релизов из мира Open Source. В начале апреля увидел свет *Rhythmbox 0.10*, флагман музыкальных плейеров Gnome, и несмотря на низкий номер версии, многие Gnome-любы нашли его очень стабильным и полезным. Версия 0.10 главным образом исправляет ошибки, ликвидировав несколько глюков и проблем с памятью, но улучшены также функции совместимости с *Jamendo*, *Magnatune* и *AudioScrobbler*.

В разделе **Рабочий стол/Rhythmbox** на нашем DVD вашему вниманию предлагается исходный код версии 0.10, который вы можете скопировать в домашний каталог и сделать следующее:

```
tar xfvj rhythmbox-0.10.0.tar.bz2
cd rhythmbox-0.10.0
./configure
make && make install
```

Помните, что последняя строка вводится от имени администратора. Если на стадии **./configure** появятся сообщения об ошибках, придется проверять, установлены ли у вас все необходимые зависимости для создания программы. Например, если скрипт configure жалуется, что не нашел *GTK*, подключите менеджер пакетов и установите библиотеки разработчика *GTK* (обычно это *GTK-devel* или *libGTK-dev*). Если вы установите пакеты разработки GCC, *GTK* и Gnome, то у вас все должно быть хорошо.

Собранный *Rhythmbox* будет установлен в директорию **/usr/local**, и вы сможете запускать его, вводя

```
/usr/local/bin/rhythmbox
```

Также на фронте мультимедиа у нас имеется *Kino 1.0.0*, первый полный релиз этого цифрового видеоредактора, чья популярность постоянно увеличивается. *Kino* дает возможность работать с материалом, отснятым различными цифровыми видеокамерами, и объединять полученные результаты с существующими AVI и DV-файлами. Если вы уже являетесь читателем *Linux Format*, вы, возможно, помните наше руководство для начинающих в **LXF80**, в котором рассказывалось, как добавлять музыку и дикторский текст в видео с помощью *Kino*.

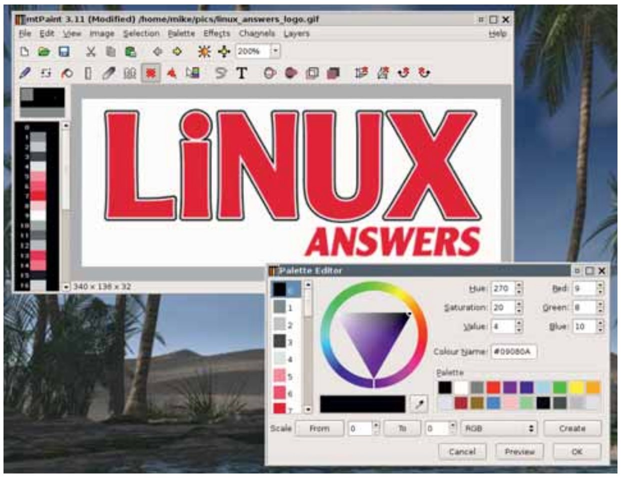
То была версия 0.8.0 – и если с ней у вас имели место проблемы, попробуйте новый усовершенствованный релиз 1.0.0, в нем исправлена куча ошибок. Среди самых важных новых функций – выгрузка на **Blip.tv**, подключение и поддержка различных съемных USB-устройств; берите исходный код из нашей директории **Рабочий стол/Kino** – и за работу.

Шестнадцатеричная утеха

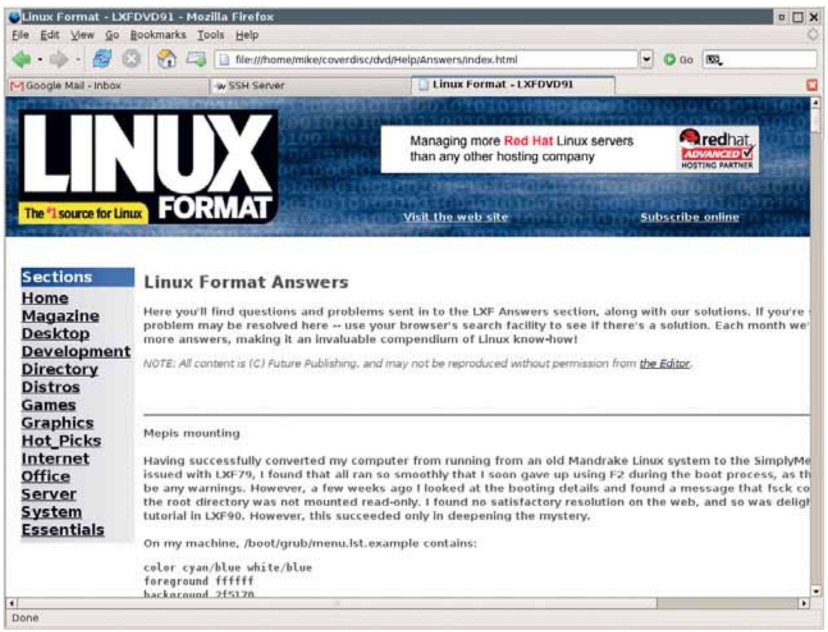
А вот кое-что для программистов: *Bless*, аккуратный шестнадцатеричный редактор, написанный на Mono. *Bless* снабжен интерфейсом со вкладками для редактирования нескольких документов, и, что самое важное, сменными цветами в основной панели редактирования данных. После получасового созерцания псевдослучайного списка чисел в надежде вскрыть формат или выяснить, где хранятся данные в исполняемом файле, вы – мы в этом уверены – по достоинству оцените любую попытку сделать интерфейс более удобным для глаз. Код ищите в разделе **Рабочий стол/Bless**.

По части графики мы предлагаем новый релиз *MtPaint*, небольшого пиксельного редактора от Марка Тайлера [Mark Tyler]. Каждый пользователь Windows (бывший или настоящий) знает *Paint*, супер-простой редактор изображений, включенный в дистрибутив ОС от Microsoft; *MtPaint* применяет тот же подход, и в нем гораздо проще разобраться, чем в мощном *GIMP*, но зато и функций гораздо меньше. Однако в плане требований он весьма скромен, и может работать на старых 200-МГц машинах с 32 МБ ОЗУ (его любят нетребовательные дистрибутивы, например, DSL). В разделе **Графика/mtPaint** на DVD вы найдете релиз 3.11 и руководство пользователя.

Стоит также обратить внимание и на *Qtm 0.5* в разделе Интернет – это отличный клиент блогов для KDE; и на самые свежие версии *MySQL* и *Apache*, которые порадуют ваш сервер; и на коммерческие драйвера для видеокарт ATI и Nvidia в разделе **Главное**. Наконец, зайдите на страничку Ответы в директории **Help**, где мы решили уже более 80 проблем с Open Source.



› **По-вашему, GIMP слишком наворочендля большинства работ? Попробуйте *MtPaint*, компактную и нетребовательную альтернативу.**



› **Решения самых частых закавык Linux – в разделе Справка/Ответы.**

И наконец...



Наработавшись с серьезными программами корпоративного уровня, вы, скорее всего, захотите расслабиться с помощью нескольких сокровищ из нашего раздела **Игры**. *Trip on the Funny Boat* (Путешествие на Забавном Корабле) потребует от вас моряцкого подхода к жанру стрелялок: вы управляете маленьким кораблем, который качается на океанских волнах и расстреливает из пушек все на своем пути. Вражеские корабли, акулы, мины –продолжите сами: все, что шевелится. Чтобы запустить игру, извлеките **funnyboat-1.5-src.zip** и запустите **python main.py** в появившейся директории.

Если вы ищете нечто более умственное, примитесь за *PyTraffic*. Это – отличная 3D-версия игры *Rush Hour* (Час Пик), в которой вам нужно собрать красную машину, перетаскивая автомобильные части по игровому полю. Как все хорошие головоломки, игра в большей степени требует способности к планированию, нежели буйной фантазии.



› **Ррр! Поживите-ка жизнью пирата в *Funny Boat* – но только без попугаев, цынги или отсутствующих конечностей.**

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК!

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Содержание LINUX FORMAT DVD Июнь 2007

Сторона 1

Рабочий стол
Bless - hex-редактор
Kamion - утилита для создания резервных копий
Kino - редактор нелинейного монтажа видео
Rhythmbox - аудиоплеер для среды Gnome
Scribus - открытая система верстки

Разработка
GDC - GCC-интерфейс для языка D
MonoDevelop - среда разработки для Mono
PHP - скриптовый язык для веб-разработки
Subversion - современная система контроля версий

Дистрибутивы
Puppy Linux - компактный LiveCD для слабых машин
StartCom Enterprise Linux 5 - свободная сборка RHEL
ChouXiang - абстрактная стрелялка
KardsGT - пакет карточных игр
PyTraffic - компьютерная версия головоломки "Трафик"

Графика
Blender - популярный открытый 3D-редактор
mtPaint - редактор для пиксель-арта

Интернет
Nvu - визуальный HTML-редактор
QTM - клиент для блогов
XChat - IRC-клиент для Linux

Офис
OpenOffice.org - популярный офисный пакет

Сервер
Apache - самый популярный веб-сервер в мире
DenyHosts - утилита для автоматической блокировки хостов
MySQL - открый сервер баз данных

Система
BackupPC - комплекс утилит для создания резервных копий
DBAN - загрузочный диск для безопасного удаления данных
fwlogwatch - это анализатор логов пакетного фильтра
Hamachi - приложение для конфигурации VPN
gHamachi - интерфейс для Hamachi
GParted - визуальный редактор разделов диска
InitNG - новая система инициализации
Lire - система анализа логов
QEMU - свободный эмулятор процессора и виртуализатор
Sentry_Tools - комплект сервисов безопасности для Unix
Squidalyser - анализатор трафика Squid
SystemImager - программа для автоматизации установки Linux

Сторона 2

Vector Linux 5.8 - дистрибутив на базе Slackware
TeX Live 2007 - издательская система LaTeX
Eclipse - платформа для построения сред разработки

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15

Поставляется вместе с журналом LINUXFORMAT номер 6(93) Июнь 2007

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Содержание DVD

ЖУРНАЛ

Blender .... Файлы из руководства по Blender.
Gtk .... Исходный код простого текстового редактора
Java EE.................... Примеры статьи и библиотеки.
Mono........................ Код примеров статьи.
PDFs .... Статьи из предыдущих номеров Linux Format.
Roundup................... Командные интерпретаторы.
WebDev .... Исходный код для учебника по разработке web-приложений.
OOo .... Практикум по *OpenOffice.org Writer*

РАБОЧИЙ СТОЛ

Bless........................ Шестнадцатеричный редактор.
Kamion..................... Инструмент резервного копирования и миграции пользователей.
Kino .... Пакет для редактирования видео.
Rhythmbox................ Музыкальный менеджер и проигрыватель.
Scribus..................... Настольная издательская система.

РАЗРАБОТКА

GDC .... Поддержка языка D для GCC.
MonoDevelop........... Интегрированная среда разработки C# .
PHP .... Язык программирования сценариев.
Subversion............... Система управления версиями.

ДИСТРИБУТИВЫ

StartCom.................. Корпоративный дистрибутив на исходном коде Red Hat.
Puppy Linux............. Оригинальный легковесный LiveCD.

ИГРЫ

ChouXiang .... Абстрактная стрелялка.
Funny Boat............... Стрелялка во все стороны.
KardsGT................... Карточная игра.
PyTraffic .... Автомобильная головоломка.

ГРАФИКА

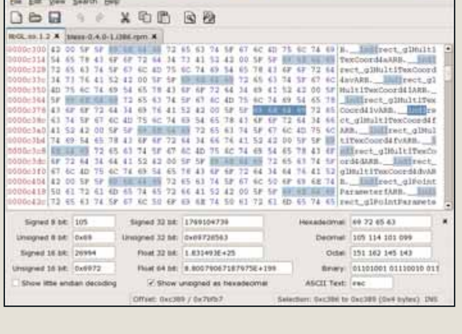
Blender .... Инструмент 3D-моделирования.
MtPaint .... Простой графический редактор.

СПРАВКА

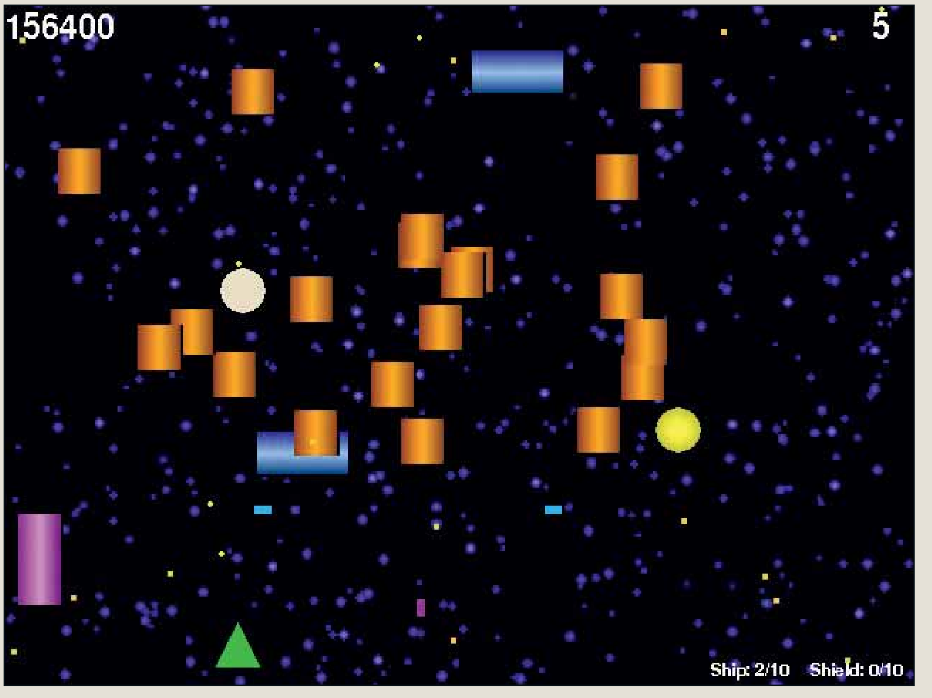
Answers................... Решенные проблемы Linux.
Rute .... Руководство администратора Linux.

HOTPICKS

Anjuta .... Интегрированная среда разработки.



› Шестнадцатеричный редактор *Bless* (см. Рабочий стол) **упрощает жизнь.**



› Абстрактных фигур-мишеней *PyGame* в игре *ChouXiang* полно.

Cedar Backup .... Инструмент резервного копирования.
Gambas.................... IDE для языка Basic.
Hinversi.................... Reversi-подобная игра.
Kflickr .... Загрузчик фотографий для KDE.
KRename .... Программа переименования файлов для KDE.
Little Polish.............. Игра в бразильские шашки.
Lm_sensors............. Инструмент мониторинга оборудования.
RecordMyDesktop .... Программа записи сессий рабочего стола.
Virgil .... Язык программирования для микроконтроллеров.

ИНТЕРНЕТ

Nvu .... Web-редактор в стиле WYSIWYG.
QTM.......................... Клиент блогов для KDE.
XChat....................... IRC-клиент.

ОФИС

OpenOffice.org .... Офисный пакет.

СЕРВЕР

Analog...................... Анализатор файлов журналов.
Apache..................... Самый популярный web-сервер в мире.
DenyHosts .... Защита от атак на SSH-сервер.
MySQL .... Сервер баз данных.

СИСТЕМА

BackupPC .... Система резервного копирования
Dban .... Darik's Boot and Nuke: спасательный CD.
fwlogwatch............... Фильтр пакетов, брандмауэр и анализатор файлов журналов.
GHamachi .... GTK-интерфейс для Hamachi.
GParted.................... Редактор разделов для Gnome.
Hamachi................... VPN-приложение.
InitNG....................... Скрипты начальной загрузки с распараллеливанием.
Lire .... Анализатор файлов журналов.
Qemu .... Эмулятор ПК.
Sentry Tools .... Локальная система безопасности.
Squidalyser.............. Анализатор трафика Squid.
SystemImager.......... Автоматическая установка Linux.

ГЛАВНОЕ

ATI driver.................. Графический драйвер.
Bash......................... Командный интерпретатор.
Cairo .... Библиотека двумерной графики.
CheckInstall .... Программа для создания двоичных пакетов.
Coreutils .... Утилиты командной строки.
GLib .... Низкоуровневая библиотека.
GTK .... Графический инструментарий.
HardInfo................... Системная информация и анализ.
Jigdo......................... Создатель ISO-образов.
Kernel .... Свежий релиз ядра Linux.
LibXML2 .... XML-анализатор и набор инструментов.
Ncurses.................... Оконный инструментарий текстового режима.
Nvidia driver .... Графический драйвер.
Python...................... Язык программирования.
Rawrite .... Программа записи образов на дискеты.
SBM .... The Smart Boot Manager.
SDL .... Библиотека мультимедиа.

Лучшие новинки открытого ПО на планете

LXF HotPicks



Ник Вейч

Чудо-математик, фанат Flickr, иногда редактор, Ник из высокой башни LXF орлиным взором оглядывает поле свободных программ в поисках симпатичных

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Kflickr » RecordMyDesktop » Anjuta » Virgil » Cedar Backup » Little Polish » Hinversi » KRename » Lm_sensors » Gambas

Загрузка фотографий на сервер

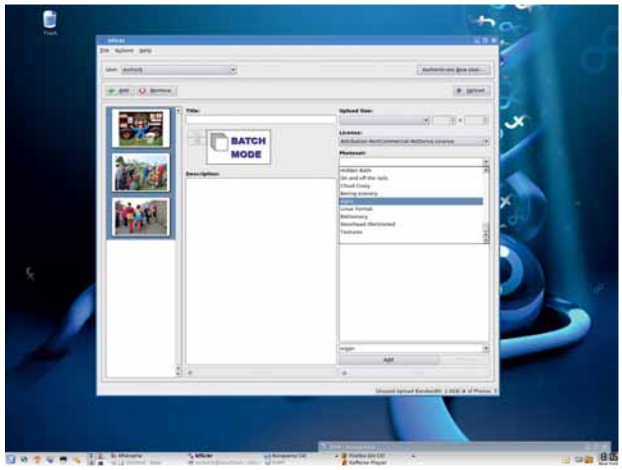
Kflickr 0.8

Версия 0.8 **Сайт** http://kflickr.sourceforge.net

Жизнь коротка – время дорого. А когда вы открываете Flickr, время становится еще более драгоценным: каждая минута, потраченная на что-то другое – это минута, за которую вы могли бы, например, написать глумливые комментарии к изображениям, загруженным кем-то, кого вы в глаза не видели. Ведь всегда приятно найти повод оторваться от нудных дел и скоренько повредничать.

Flickr хорош всем, но его встроенному загрузчику изображений призов не видать. Он основан на web, весьма медлителен и способен одновременно обрабатывать лишь пять изображений. Плюс нужно дожидаться загрузки изображения, прежде чем вводить соответствующие заголовки, комментарии и тому подобное. Тоска! Однако создатели Flickr сделали доброе дело: опубликовали API для манипуляции с системой Flickr. В результате имеется множество способов закачивать изображения, в том числе и из приложений вроде *F-Spot*.

Хотя загрузка изображений вашим менеджером фотографий выглядит проще, иногда можно наткнуться на ограничения: множество встроенных загрузчиков изображений справляются с задачей, но не выполняют всех, возможно необходимых вам, специфичных для Flickr шагов, вроде установки уровней приватности и лицензирования или добавления фотографий к готовому набору. Конечно, не всем пользователям они нужны, но если вы из таких, то зачастую отдельный пакет для закачки – это лучший вариант, а лучший из них для Linux – *Kflickr*. Как вы поняли, это KDE-приложение. Предыдущие версии включали простейшие операции над графикой и позволяли перетаскивать изображения затопленных велосипедов (у каждого своя любимая тема, верно?) в главное окно для загрузки. Появлялся список миниатюр (вы могли изменять порядок, перетаскивая их). Выбрав отдельное фото, вы меняли его индивидуальные свойства или выбирали группу, чтобы навесить общие тэги и т.п.



› **Загрузка, добавка тэгов, имен и комментариев – все сразу.**

В последней версии 0.8 слегка изменилась раскладка, сделав все чуть более доступным. Левая область содержит панель для переноса на нее изображений. В центральной части имеются области для ввода тэгов, комментариев, имен, размеров и изменения лицензии (хотя нам бы хотелось, чтобы здесь по умолчанию применялись настройки Flickr или была возможность сохранить параметры).

Функция автозавершения тэгов сберегает кучу времени – приложение запоминает все когда-либо введенные вами тэги и пытается дописать их за вас. Хотя это не всегда удобно. Если, к примеру, имея тэг 'flower', вы пытаетесь ввести 'flow', то он автоматически дополнится до 'flower', если вы не нажмете в конце пробел, что слегка раздражает.

При всем при том, *Kflickr* интуитивно понятен и, несомненно, облегчает бремя загрузки больших групп файлов. Ура!

Изучаем интерфейс Kflickr

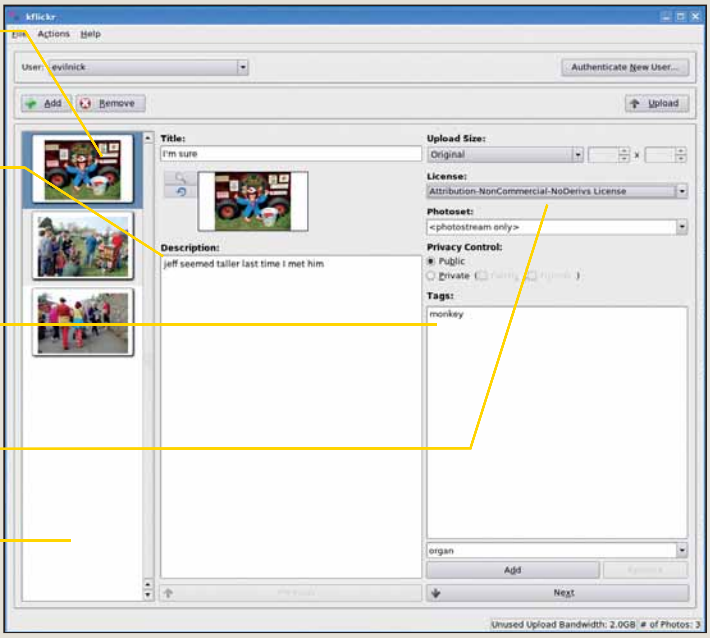
Список фотографий
Перенесите сюда ваши фото, чтобы начать их загрузку в вашу фотогалерею Flickr.

Поле описания
Эта текстовая область предоставляет достаточно места для детального описания

Добавьте тэги к фото
С удобной/навязчивой функцией автозавершения.

Подробности
Настройте остальные параметры

Пакетная обработка
Выберите несколько изображений из списка для пакетной обработки.



Запись действий с рабочего стола

RecordMyDesktop

Версия 0.3.4 **Сайт** http://recordmydesktop.sourceforge.net



› Он работает! Он реально работает! Реально!

Порой надо верить мечте. Однажды команда *Linux Format* размечталась о мире, где легко создавать видеоролики, обучающие работе с приложениями, накладывать голос (увы, Орсон Уэллс [Orson Welles] уже не с нами, но Грэм послужит терпимой альтернативой) и нежиться в ореоле похвал читателей за еще одну прекрасно выполненную работу. Бесконечные упражнения с различными схемами, программами, удаленными рабочими столами и странными внешними устройствами забили мечту практически до смерти. Но... Смотри-ка! Что это? *RecordMyDesktop*? Гммм, похоже, что и вправду можно записывать все происходящее на нашем рабочем столе, и даже с достаточной скоростью, а также в удобном формате, который мы сможем использовать. Небось, фокус какой-нибудь!

Нет, милые читатели, уверяем вас, это вовсе не фокус. Поверьте, программа работает. Эта жемчужинка честно записывает все, что творится на рабочем столе, с приличной скоростью (естественно, на нее влияют размер экрана, память и мощности процессора, но вы в любом случае будете удивлены), в формате Ogg Theora. Если у вас достаточно памяти, то он сначала буферизует запись, а сжимает ее в конце, в противном случае повышенная нагрузка не позволит получить те 25 кадров в секунду, что получили мы.

Нужно всего лишь указать файл для сброса видео, и вперед:

```
RecordMyDesktop ploppy.ogg
```

Как сказано выше, по умолчанию действие сводится к кэшированию и записи сырых данных, а затем, по завершении, они кодируются, но при недостатке места на диске вы можете использовать ключ **--on-the-fly-encoding**. Также можно задать смещение по оси X и Y, высоту и ширину или захватить отдельное окно на рабочем столе, вот так:

> «Эта жемчужинка честно записывает, что творится на рабочем столе.»

```
RecordMyDesktop --on-the-fly-encode --x 112
--y 100 --width 800 --height 600 evil.ogg
```

Итак, мы сказали, что *RecordMyDesktop* работает, и он работает – практически во всем. Для записи окон он широко использует *XDamage*, а значит, записывает изменения, а не кадры целиком. Это почти всегда работает прекрасно, но кое-что записать нельзя, например, видео, воспроизводящееся Xv (на большинстве проигрывателей, таких как *MPlayer*, режим просмотра легко переключить), или некоторые 3D-игры. Обходной вариант – использовать режим **--full-shots**. Он решит проблему, но, естественно, съест большую часть процессорной мощности.

IDE для Gnome

Anjuta

Версия 2.1.2 **Сайт** http://anjuta.sourceforge.net

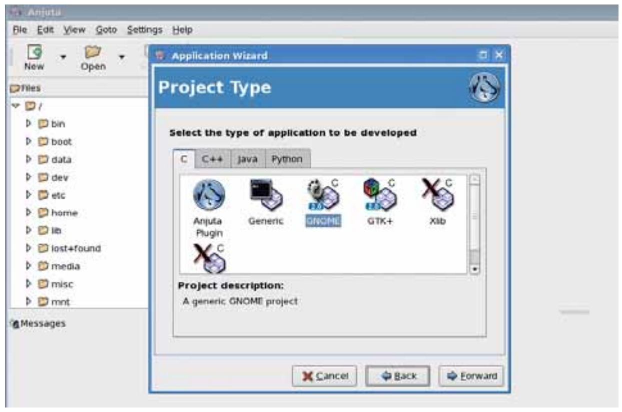
Считается, что Linux прекрасно оснащен инструментами для написания кода. В конце концов, люди каждый день только этим и занимаются, верно? Но если копнуть, то в Linux не так уж много интегрированных сред разработки топ-класса, возможно, потому, что многие разработчики засиделись в *Emacs* и *Vim*. Среди небольшого числа IDE, *KDevelop* во многих отношениях превосходит остальные, но жаждущие полноценной, полнофункциональной IDE для разработки Gnome-приложений, вероятно, будут более чем счастливы с *Anjuta*.

Рассматриваемая здесь версия – последняя бета 2.1.2, но в ней налицо значительные улучшения в работе внешних инструментов, профилировании, поддержке языков и общем администрировании проектов, наряду с ликвидацией множества ошибок. При создании графического интерфейса Gnome-приложений *Anjuta* хорошо работает с *Glade*, но вы, разумеется, можете также создать прочие C/C++ проекты; поддерживаются и другие языки, особенно Java и Python.

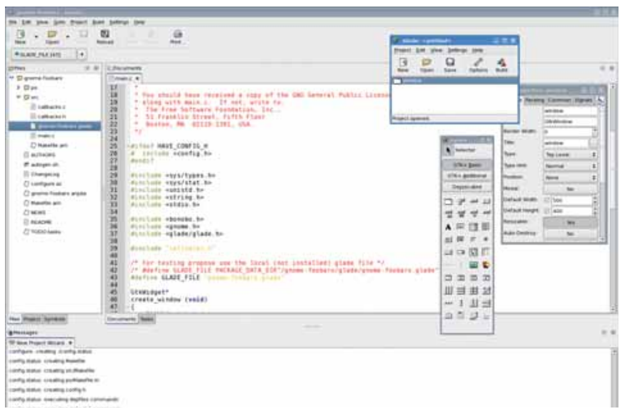
Создание приложений упрощено благодаря использованию прекрасного мастера проектов, который на предусмотренном примере кода проведет вас через настройку всех файлов и настроит *AutoGen*. Для продвинутых пользователей, вероятно, лучшей функцией этой бета-версии будет улучшенное профилирование и отладка.

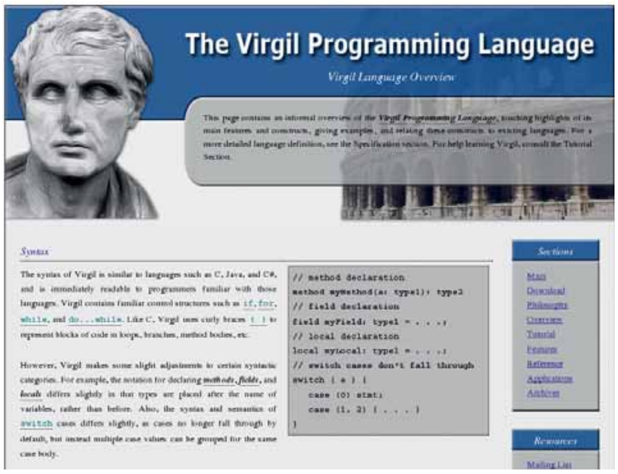
Интеграция с *Glade* действительно упрощает создание *GTK*-приложений, а запуск проходит с минимумом суеты. Однако большое разочарование вызывает отсутствие в *Anjuta* качественной документации (а иногда документации вообще), и если вы хотите помочь проекту, вот вам поле для деятельности.

Для опробования новых функций *Anjuta* потребуются последние версии *gnome-build* и *gdl* (библиотеки разработки Gnome). Но если вы серьезно разрабатываете приложения Gnome, то они наверняка у вас уже установлены!



› Создание проектов GNOME на C++ при помощи Anjuta.





Компилятор для микроконтроллеров

Virgil

Версия B-02 **Сайт** compilers.cs.ucla.edu/virgil

Поклонники *Энеиды*, радуйтесь. А может, и нет. Хотя этот язык программирования назван в честь Публия Вергилия Марона, автора древнеримской поэмы, с героическим пентаметром у него мало общего. Да, мы сказали «язык программирования». Читатели, отшвырнувшие эту заметку, вероятно, пробурчали себе под нос, что новый язык программирования нужен миру, как еще одна теплоэлектростанция. Но у Virgil есть фишка, поверьте. Это не просто еще один язык программирования для Linux: он предназначен для микроконтроллеров.

Устройства вроде PIC-процессоров от Microchip и AVR-контроллеров от Amtel являются сердцем множества крошечных гаджетов и самодельных устройств, грозящих заполнить весь мир макетными платами и запчастями от старых принтеров. Цель этого проекта – облегчить большие работы.

Синтаксически язык очень похож на C или Java, и если вы знакомы с ними, то велика вероятность, что вы сможете читать Virgil-код без частого обращения к онлайн-руководству. Большая часть кода группируется в «компоненты», то есть полнофункциональные блоки, содержащие определения переменных и методов. Классы тоже поддерживаются, так что язык не такой уж непривычный.

Язык строго типизирован, имеет *int*, *char* и *bools*, но также добавлен низкоуровневый тип для улучшенной манипуляции с данными на уровне битов. Он эффективен при использовании часто ограниченного пространства или экзотических величин, которые бывают нужны во встроенных приложениях. Например:

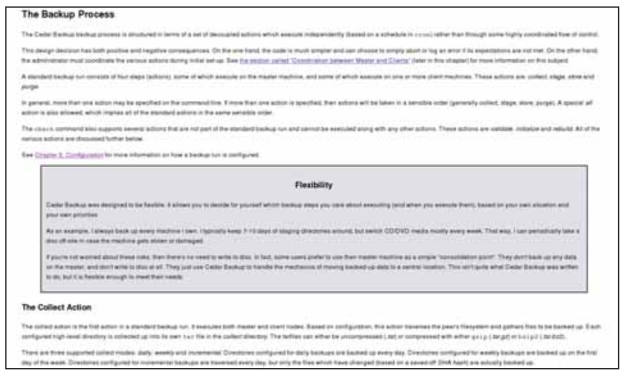
```
component RawBits{
  field outflags: 6;
  field adcdata: 12;
  field tristate: 3 = 0b101;
  field outdata: 32 = 0xffff0001;
  method getBit( a: 32, b: int): 1{
    return a[b];
  }
}
```

> «Микроконтроллеры — это сердце множества крошечных гаджетов.»

› Пусть это не похоже на стихи, но оно заставляет контроллеры AVR петь.

Этот простой компонент определяет некоторый низкоуровневый тип данных и метод, возвращающий значение указанного бита. В исходных текстах содержится множество примеров, но чтобы компилировать программы для устройств Atmel, не помешает заиметь avr-gcc.

Хотя для микроконтроллеров есть и высокоуровневые языки типа C, для некоторых из них подобная абстракция столь далека, что теряются тонкости, обуславливающие эффективность работы устройства. С Virgil все не так.



Система резервирования

Cedar Backup

Версия 2.13.1 **Сайт** http://cedar-solutions.com/software/cedar-backup

Никто этого не любит, но делать это иногда приходится. Нет, не благодарить Майка за кропотливую работу над диском-приложением: мы имели в виду резервирование важных файлов, чтобы с ними не случилось беды. В Linux имеется много решений для резервирования, и мы уже публиковали описания и учебники по таким инструментам, как *Amanda*.

Cedar Backup мы не затрагивали, но оно может быть интересно пользователям с парой машин в сети (или без нее). *Cedar Backup* – просто набор скриптов, выполняющих резервирование на CD или DVD, с различными схемами архивирования и опциями частичной автоматизации. Приложение *Cedar* также знает, как резервировать различные типы данных, например, из базы *MySQL*, и это вселяет уверенность, что при необходимости вы всегда сможете восстановить данные.

Установка приложения достаточно проста: оно написано на Python и имеет мало зависимостей, так что проблемы маловероятны. А вот настройка, надо признать, сложновата.

Процесс конфигурирования состоит в создании «набора» данных для резервирования, затем настройки клиента и главного узла. В случае одной машины, ясно, что клиент и главный узел одно и то же, но вам все равно необходимо будет их настроить. Вам также придется создать дополнительного пользователя для запуска программы – некоторые дистрибутивы уже имеют пользователя «backup», и вы всегда можете его использовать.

Мы не будем описывать настройку полностью, потому что она сильно зависит от вашей конфигурации. Причина, по которой мы о ней упомянули – в отличие от множества других открытых проектов, эта система резервирования имеет детальную и обширную документацию. И ей легко следовать. Заметим все-таки, что стандартная процедура оценки резервирования рекомендует не только резервировать, но и убедиться, что вы можете восстановить данные с резервной копии! Вероятно, на одной машине труднее делать это и все еще быть уверенным, что ваши данные в безопасности, но по крайней мере следует проверить созданные резервные копии, чтобы убедиться, что сохранено именно то, что нужно – сделайте это сейчас, потом может быть поздно.

> «Оно знает, как резервировать разные типы данных, например, MySQL.»

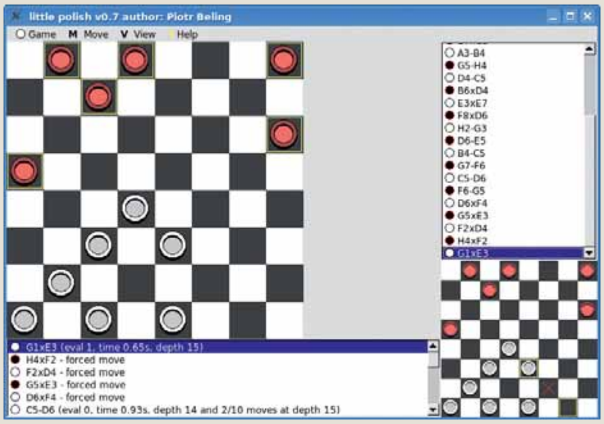
› Приложение для резервирования – с хорошей документацией? Ух ты!

HotGames Развлекательные приложения

Бразильские шашки

Little Polish

Версия 0.7 **Сайт** http://warcaby.w8.pl



› Играйте в международные польско-бразильские шашки!

В этом месяце в HotGames включены настольные игры. Стрелялки от первого лица и платформы, переполненные лазерами, плюющимися огнем – это хорошо, но надо признать, что хитрость и стратегия – тоже часть игрового процесса, и мало какие игры интеллектуальнее шашек. Да, верно: все фигуры одинаковы, используется лишь половина клеток и есть фуки и дамки, и так далее.

Не кривитесь заранее: обычные шашки мы бы предлагать не осмелились. Нет, это бразильские шашки. Вы можете спросить: а чем они отличаются от классических английских, которые вы учили на коленях у дедушки? Ну, они похожи на международные, но играются на доске 8x8, как и английские. Международные шашки? Это то же самое, что и польские (видимо, отсюда и название). Все еще неясно? Короче, игра весьма похожа на ту, что вы уже знаете (конечно, если вы не поляк), но, что важно, шашкам разрешено делать ход назад, если при этом берется шашка противника. На эту небольшую уловку многие попадались не один раз, и она ведет к неумолимому массовому уничтожению.

Игра весьма прямолинейна: компьютер подсвечивает шашки, которые могут ходить. Щелкните на одной, затем щелкните на одном из предложенных ходов. Но ситуации с множественным взятием и «фуками» слегка ошарашивают, когда компьютер автоматически передвигает шашки за вас. В играх с большим количеством взаимозависимых ходов это выглядит так, словно кто-то подошел и смахнул половину шашек; но все честно происходит над доской.

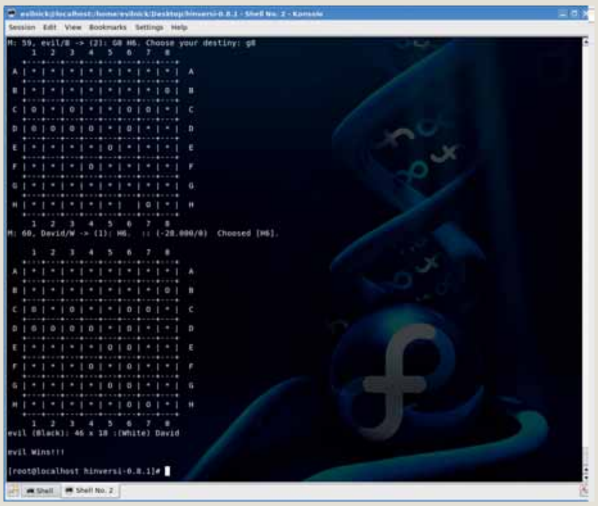
> «На эту уловку многие попадались не один раз...»

Здесь, в Башнях **LXF**, эта искусная маленькая игра смогла победить даже наших передовых шашистов, так что мы думаем, что она обеспечит вам хорошую игру. Если вы победите ее, напишите и дайте нам знать (letters@linuxformat.ru)!

Реверси

Hinversi

Версия 0.8.1 **Сайт** http://freshmeat.net/redir/hinversi/47919/url_homepage/hinversi.html



› Поверните экран на 90°, и увидите бледное бородатое лицо Артаксеркса. Жуть!

Древнюю игру реверси изобрели в древней же Персии. Дата первой партии неизвестна, но игра была весьма популярна, и в ней использовались камешки, добываемые на одной из трех специальных каменоломен. Играли в нее и правители – по некоторым свидетельствам, один «царь царей» играл на огромной площадке со сторонами в 500 метров, а команды, отличающиеся цветом колесниц, размещали и удаляли гигантские фишки. Ну да, мы все это наврали, но ведь как складно, правда?

На самом деле, реверси была придумана англичанами в 1880-х, и не обрела всемирной популярности, пока не вышла под названием Othello в 1970 г. в Японии. Если вы знаете больше, поправьте нас.

Откуда бы она ни взялась, реверси завоевала некоторую популярность на компьютерах, потому что описывается набором простых правил, следуя которым, легко записать алгоритм моделирования игры и поиска наилучшего хода. Данная версия многим обязана играм, возникшим на заре Unix, когда в этой сфере доминировали бородатые чудаки. Почти как сегодня.

Так или иначе, *Hinversi* – хорошая игра, требующая минимума ресурсов, и мы можем уверенно предсказать, что она запустится на любой машине в любом дистрибутиве – во многом благодаря работе в консоли. Для вас это проблема? Что ж, если астронавты на Аполлоне могли играть в шахматы без доски, то вы уж наверняка сумеете сыграть в реверси при помощи '*' and '0'.

> «Следуя простым правилам, записать алгоритм игрового процесса легко.»

Новая версия стала играть лучше, но, как видно на рисунке, все еще пасует перед нечеловеческой мощью загадочного игрока под доходчивым именем Зло...

Утилита рабочего стола

KRename

Версия 3.0.14 **Сайт** www.krename.net

Странно, что стандартный рабочий стол Linux не имеет простого способа переименования файлов. Да, это делается вызовом свойств иконки правым щелчком, либо нажатием F2 в *Konqueror*. Но если нужно переименовать их сотню? Это происходит постоянно, в основном с файлами изображений – бесконечные файлы 'IMG001.jpg' хочется назвать как-то осмысленно, возможно, включив в имя дату или место создания. Или нужно загрузить набор конфликтующих имен в резервную копию. Или... короче, вы поняли.

KRename, вероятно, лучшая и хорошо известная попытка исправить это недоразумение. При запуске приложения возникает простое окно с четырьмя вкладками и несколькими кнопками по нижнему краю. Перенесите группу файлов в окно или выберите их при помощи диалога. На последней вкладке выбирается шаблон для переименования.

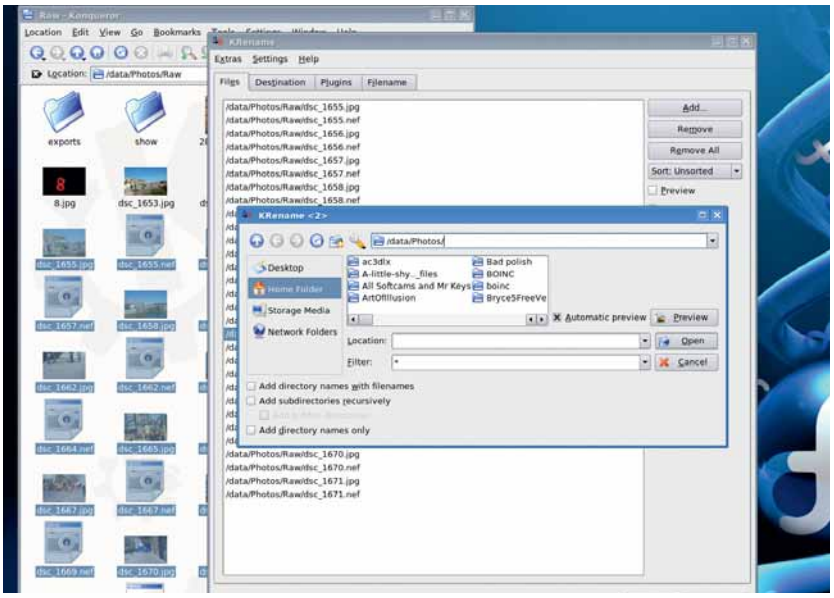
Наверняка вы захотите использовать часть исходного имени – для этого просто щелкните на кнопке Insert Part Of Filename [Вставить часть имени файла]. На экране появятся примеры имен, а вы выберите что понравилось, и шаблон готов. Разумеется, вы можете создать собственные шаблоны, если хотите переупорядочить ваши файлы по названиям созвездий или как-либо еще: хозяин – барин. Осуществляйте выбор и замену определенных частей имен, или переупорядочивайте числа с начала, если желаете – все предусмотрено.



› **Выберите желаемый шаблон файла.**

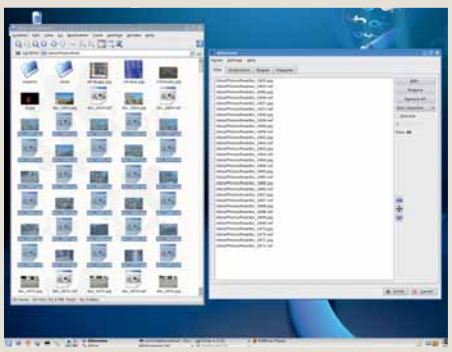
Другие вкладки отвечают за дополнительные операции. Вас, вероятно, заинтересует размещение. По умолчанию файлы просто переименовываются там, где находятся, но вы можете скопировать или переместить их еще куда-нибудь. Можно даже создать скрипт отмены, восстанавливающий исходные имена, хотя мы прямо и не знаем, кому это надо.

Последняя нерассмотренная вкладка – Модули. Эти компоненты могут по вашему желанию что-нибудь сделать с файлами при их переименовании или перемещении. Обычный фаворит здесь – опция изменения прав доступа, но есть и другие, менее очевидные, но не менее полезные опции: главная из них, пожалуй, модуль Command, он автоматически выполняет некоторые команды *Bash* (или их список) при переименовании файлов; это удобно для пакетного конвертирования. Возможность сохранить эти команды не только в профиле (который сохраняется всегда) – вот чего недостает *KRename*.



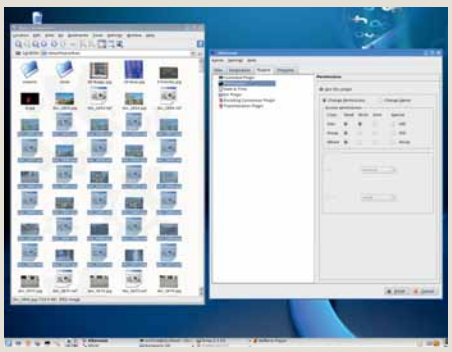
› **Пакетная обработка: ваш способ избавиться от нудной работы.**

Шаг за шагом: Переименование пакета файлов



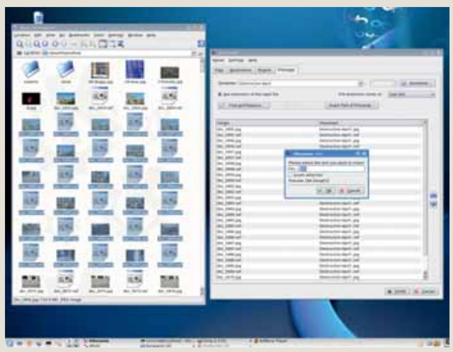
» Drag and drop

Для начала перенесите файлы в главную панель с рабочего стола или из *Konqueror* (или добавьте их посредством диалога выбора файлов)



» Выберите модули

Хотите ли вы изменить и права доступа при переименовании файлов, или выполнить некоторые преобразования? Делайте это здесь!



» Назовите и вперед

Выберите новый шаблон имени файла – может быть, вы захотите использовать часть исходного имени. Нажмите OK, а затем Finish.

Менеджер устройств

Lm-sensors

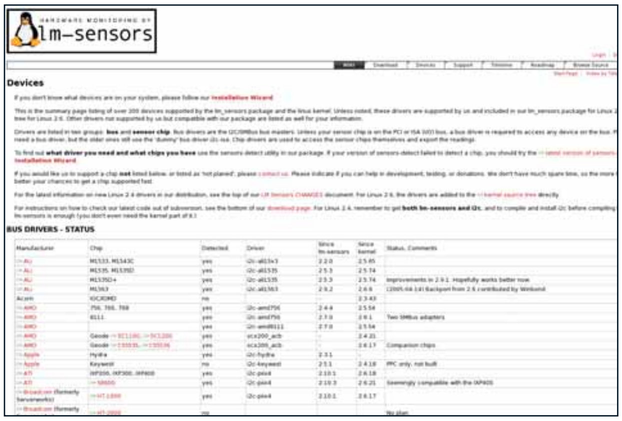
Версия 2.10.3 **Сайт** www.lm-sensors.org

Вы фанатик контроля? Все нормально, не бойтесь: вы среди друзей. Одно из изящных нововведений в компьютерах за последние лет 15 – системы мониторинга, добавленные, практически как стандарт, в чипсеты материнских плат. Среди поддерживаемых функций – контроль за температурой и вентиляторами, но есть и более экзотические. Некоторые поддерживают двунаправленный поток информации, например, позволяют ОС изменять скорость вращения вентиляторов.

Однако все они жестко связаны с чипсетом, встроенным в материнскую плату, и Linux частенько не входит в список ОС, поддерживаемых изготовителем. Для тех из нас, кто требователен к своей ОС, имеется прекрасное решение: проект *Lm-sensors* (Грэм описывал его в номере LXF92).

Эти драйвера, вообще-то, уже включены в ядро, так что для подключения функций мониторинга в своем настольном ПК или на сервере вам в большинстве случаев ничего не нужно делать. Но, покупая самую новую материнскую плату, помните, что проект *Lm-sensors* выпускает очередную версию по мере написания кода. Это означает, что вам придется перекомпилировать ядро (что в некоторых дистрибутивах становится весьма проблематичным); зато вы будете знать, сколько оборотов в минуту делает ваш вентилятор.

› **Следите за работой внутренностей без эндоскопа.**



Инструментарий разработки программ

Gambas

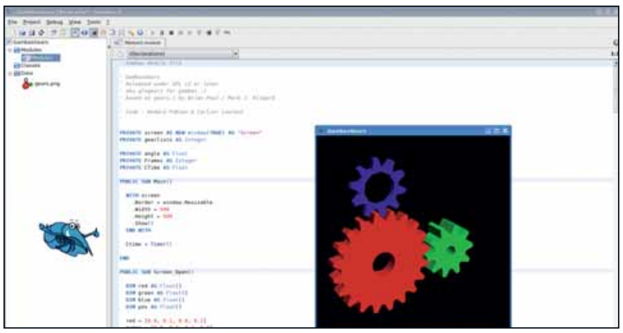
Версия 1.9.48 **Сайт** http://downloads.sourceforge.net/gambas

Gambas, кроме водного членистоногого, «почти значит» Basic. По крайней мере, так трактуют этот традиционный рекурсивный акроним его создатели. Gambas стал скандально известным сразу же после первого релиза, поскольку объявлялось, что это будет ответ Linux на Visual Basic, но оказалось, что вероятный ответ на Visual Basic все-таки C#. Тем не менее в IDE/интерпретаторе Basic немало интересного, особенно когда можно с легкостью строить графический интерфейс приложений.

Описываемая версия Gambas – еще один опытный релиз, поэтому, как обычно, кому-то подойдет, а кому-то нет. Но его стоит рассмотреть, потому что он содержит много функций, добавленных после последнего стабильного релиза: улучшенная интернационализация, множество виджетов, улучшенная подсветка синтаксиса...

После версии 1.9.48 мы, естественно, в недалеком будущем ждем полной стабильной версии 2.0, но и в текущем состоянии программа вполне готова к использованию, и мы уверены, что команде разработчиков пригодятся дополнительные отзывы. Так что вперед – исследовать. Сейчас имеются общественные проекты, охватывающие большинство вариантов применения Gambas. В какой-то степени Basic выглядит как шаг назад – если вы собрались изучить какой-нибудь полезный язык, уж лучше взяться за C/C++, C# или Java. Но если вы уже знакомы с Visual Basic, то Gambas может быть более легкой дорогой к разработке под Linux.

› **Не дайте имени обмануть вас: gambas – не морепродукт.**



Также выпущены

Новые и обновленные программы, заслуживающие вашего внимания...

» **Bless 0.5.0** Лучший шестнадцатиричный редактор для Linux http://freshmeat.net/projects/bless

» **BusyBox 1.5.0** Базовые команды в одном исполняемом файле – встраиваемый рай http://busybox.net

» **Clive 0.1.9** Получите видео с YouTube http://home.gna.org/clive

» **Doxygen 1.5.2** Выполняет тяжелую работу по документированию www.stack.nl/~dimitri/doxygen

» **HylaFax 5.1.2** Отправка/получение факсов без факс-аппарата www.hylafax.org

» **Lftp 3.5.10** Необходимый FTP-инструмент командной строки http://lftp.yar.ru



› **Параллельная передача файлов в Lftp.**

» **Libcurl 7.16.2** Незаменимый инструмент для загрузки данных из web http://curl.haxx.se

» **Linphone 1.7.0** Совместимый с SIP аудио/видео телефон для Linux www.linphone.org

» **Postfix 2.4.0** Новая стабильная версия популярного MTA www.postfix.org

» **PyQt 4.2** Обертка Python для последней версии Qt www.riverbankcomputing.co.uk/pyqt

» **Sendmail 8.14.0** Отправляет почту по назначению www.sendmail.org

» **Webmin 1.340** Наверное, единственный инструмент удаленного администрирования, который вам понадобится. www.webmin.com



› **Webmin: системное администрирование на базе web.**

» **Wine 0.9.34** Запускайте Windows-приложения без Windows. Ну, вперед! www.winehq.com LXF

Школа LINUX FORMAT

Мы продолжаем публикацию методических пособий для изучения открытых программных продуктов в образовательных учреждениях.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Сделайте это вместе с OpenOffice.org Writer

Пожарина Галина Юрьевна,
начальник отдела информационных технологий,
Санкт-Петербургский технический колледж управления и коммерции

В силу причин правового, экономического характера, а также учитывая стандарт качества программного продукта общеофисного назначения OpenOffice.org, использование его широким кругом пользователей для решения профессиональных и частных задач является оправданным.



Пакет Open Source программного обеспечения офисного назначения OpenOffice.org распространяется по лицензии GPL/LGPL.

OpenDocument format признан в качестве открытого формат стандарта OASIS и ISO стандарта офисных документов.

OpenOffice.org - кроссплатформенный, доступный, простой в установке.

Sun, IBM, HP, Red Hat, Novell ,Adobe сочли возможным огласить полную поддержку в своих продуктах и службах открытых стандартов XML Евросоюза, уже работающих в компонентах OpenOffice.org.

Развитие этого продукта не контролируется каким-либо одним разработчиком или одной организацией .

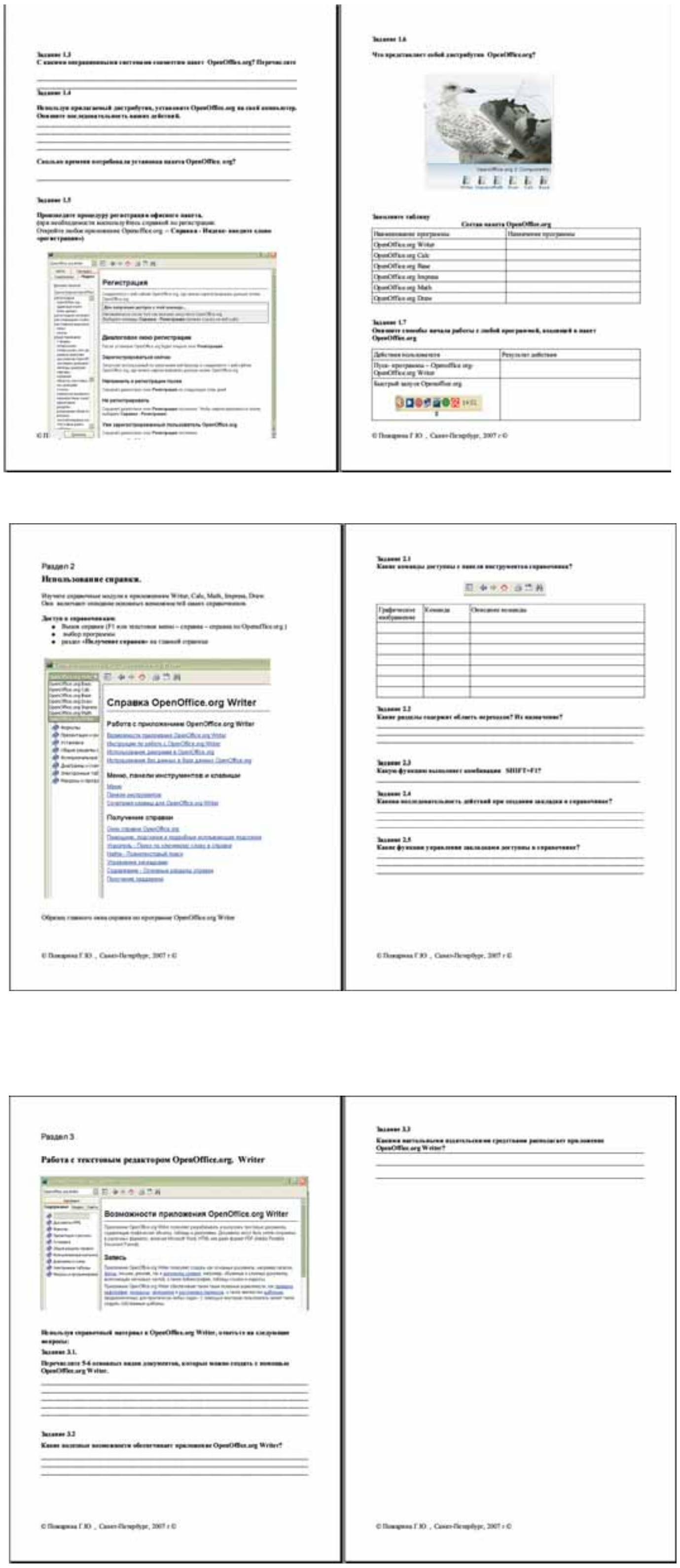
Одна из основных отличительных черт программного обеспечения на основе Open Source – динамичность и обновляемость. Частота, с которой выходят новые релизы программного обеспечения, достаточно высока. Подобная гибкость позволяет оперативно устранить неудобства функционального характера, подстроить оболочку с учетом эргономических факторов, внести необходимые коррективы в исходный код, исправив недоработки и ошибки.

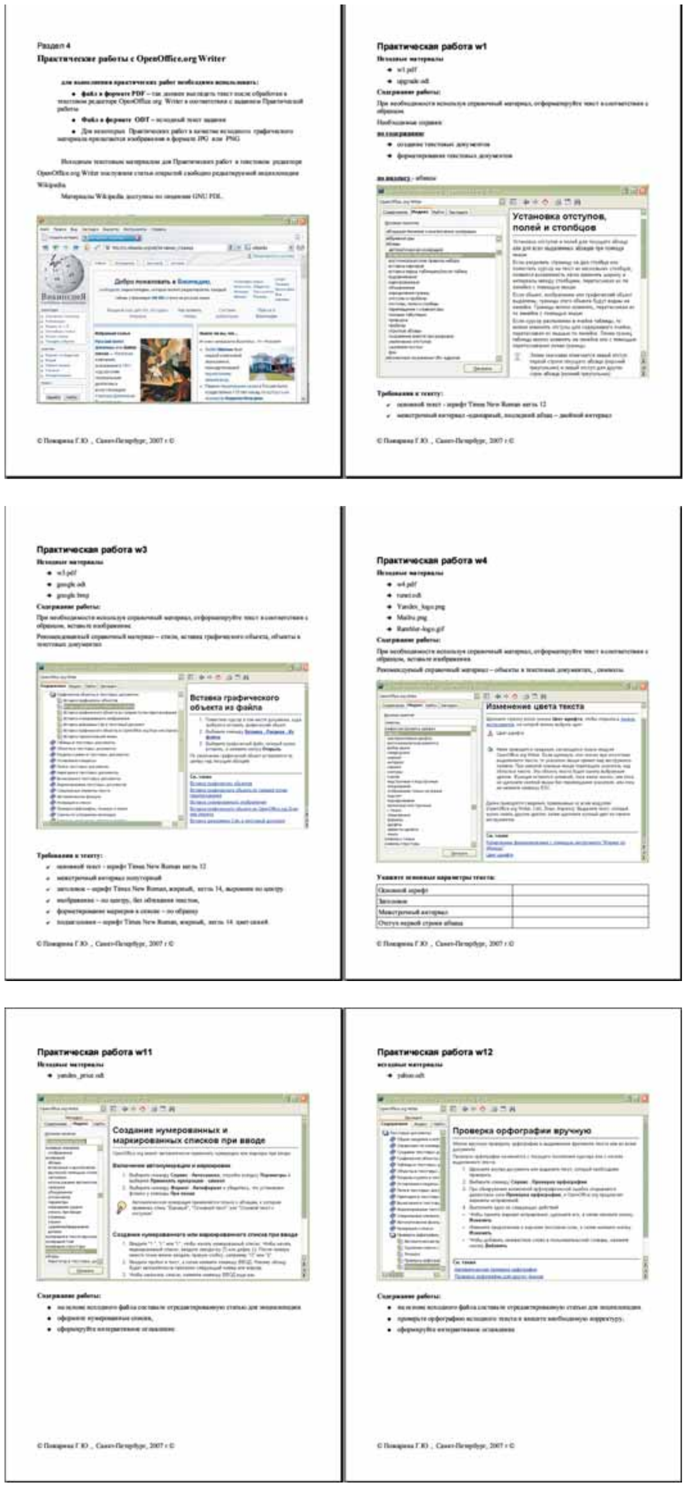
Инструкции по использованию программного продукта и описание специфики каждого релиза в виде технической документации присутствуют в файлах справки.

Ориентация в справочном материале может и должна составлять основу культуры использования программного обеспечения, особенно программных продуктов на основе Open Source. Данная особенность делает программное обеспечение на основе Open Source ценным материалом для дидактических целей.

Методички LXF

Сделайте это вместе с OpenOffice.org Writer





Это пособие вы найдете на нашем диске в файле **writer_pozharina.pdf**
Тексты, использованные в примерах, находятся в файле **texts_for_practicum.odt**

Продолжение – в следующих номерах.

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Иван Мищенко
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Валентин Развозжаев, Александр Черных, Юлия Шабунио
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Литературный редактор
Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Шарлотта Мартин (Charlotte Martyn), Мэт Нейлон (Matt Nailon)

Подготовка материалов
Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Энди Ченнел (Andy Channelle), Эмиал Ченнер (Amias Channer), Дэвид Кулсон (David Coulson), Бен Харлинг (Ben Harling), Энди Хадсон (Andy Hudson), Рейчел Проберт (Rachel Probert), Натан Сандерс (Nathan Sanders), Александр Бабаев, Евгений Балдин, Андрей Боровский, Павел Воронцов, Евгений Гаврилов, Петр Семилетов, Алексей Федорчук, Илья Шпаньков
Художественные ассистенты: Эми Кустон (Amie Causton), Анна Фишер (Anna Fisher)
Фотографии: Джена Камбо (Jena Cumbo), Джейсон Каплан (Jason Kaplan)
Иллюстрации: Нейл Бартлет (Neil Bartlett), Пол Блехфорд (Paul Blachford), Крис Винн (Chris Winn)

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Представительство в Москве:
пр.Мира, 161, телефон +7(495) 799-18-63, +7(495)136-88-45
Email: info@linuxformat.ru, Web: www.linuxformat.ru

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO •SAN FRANCISCO



в июле

Дистрибутивы на ринге

Взвесим все «за» и «против»: поддержку сообщества, скорость выпуска исправлений и прочее. Кто окажется победителем на этот раз?

Заставьте Wine работать

Хотите запускать любимые Windows-приложения в Linux, но не знаете, как? **Майк Сондерс** подготовил для вас специальное руководство.

Свободное ПО минус Linux

Как перевести друзей и коллег, использующих Mac/Windows, на рельсы Open Source?



Андреас Типалдос

Исполнительный директор Xandros – о том, почему Linux должен уметь работать с Windows.

Содержание последующих выпусков может меняться без уведомления

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» 6 номеров (июль, август, сентябрь, октябрь, ноябрь, декабрь 2007 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2007 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2007 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

- по выставленному счету (для юридических лиц)
- по квитанции в любом отделении Сбербанка

Плюсы подписки

- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2007!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ НА ЖУРНАЛ

Linux Format

индекс издания

количество комплектов

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА ______ почтовый индекс ______ адрес доставки

КОМУ ______ фамилия, инициалы

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ | МЕСТО | ЛИТЕР

НА ЖУРНАЛ ______ индекс издания

Linux Format

наименование издания

СТОИМОСТЬ	подписки	руб.	коп.	количество комплектов	
	переадресации	руб.	коп.		

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА ______ почтовый индекс ______ адрес доставки

КОМУ ______ фамилия, инициалы

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса» – совместный проект Государственного предприятия «Казпочта», Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан

- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения

- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия

- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия

- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан

- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия

- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул.Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан

- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина

- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22

ПОДПИСКА НА LINUX FORMAT

Агентство "Centerpress"

Сколько стоит подписка?
Подписка на журнал "Linux Format" 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь 2007 года) стоит 1800 рублей.

Как оформить подписку?
Чтобы оформить подписку на журнал "Linux Format", необходимо зарегистрироваться в интернет-агентстве Centerpress.ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар "Подписка на журнал "Linux Format" на 2007 год 12 номеров (01-12 / 2007)", получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Агентство "Centerpress" : www.centerpress.ru

Все Плюсы подписки!
- Подписка дешевле!
- Гарантированное получение журнала!

По каталогам РФ

Каталог агентства "РОСПЕЧАТЬ" - подписной индекс

20882

Каталог "ПРЕССА РОССИИ" - подписной индекс

87974



АЛЬТЕРНАТИВНЫЕ АГЕНТСТВА РФ

Агентство «Интер-Почта»
(095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса»
(095) 787-34-47

Агентство «УралПресс»

- Екатеринбург, Березовский, В. Пышма, Первоуральск
  тел. (343) 375-80-71, 375-84-93, 375-84-39,
  факс 375-62-74,
  info@ural-press.ru
- Нижний Тагил
  тел. (3435) 411448, 417709,
  ntagil@ural-press.ru
- Челябинск
  тел. (351) 262-90-03, 262-90-05,
  pochta@chel.surnet.ru
- Пермь
  тел. (3422) 60-24-40, 60-22-95, 60-35-42,
  parma-press@permonline.ru